

TUGAS AKHIR - TE 090362

PROTOTIPE SIMULASI SISTEM PENYALURAN AIR BERSIH DI DESA BERBASIS RFID

Gamal Armando
NRP 2211 030 066
M Imam Ainul Hidayat
NRP 2211 030 116

Dosen Pembimbing
Slamet Budiprayitno, ST., MT.

PROGRAM STUDI D3 TEKNIK ELEKTRO
Fakultas Teknologi Industri
Institut Teknologi Sepuluh Nopember
Surabaya 2015



FINAL PROJECT - TE 090362

***PROTOTYPE SIMULATION OF WATER DISTRIBUTION
SYSTEM BASED ON VILLAGE WITH RFID BASED***

Gamal Armando
NRP 2211 030 066
M Imam Ainul Hidayat
NRP 2211 030 116

Consellor
Slamet Budiprayitno,ST.,MT.

ELECTRICAL ENGINEERING D3 STUDY PROGRAM
Industrial Technology Faculty
Institute of Technology Sepuluh Nopember
Surabaya 2015

PROTOTIPE SIMULASI SISTEM PENYALURAN AIR BERSIH DI DESA BERBASIS RFID

TUGAS AKHIR

**Diajukan Guna Memenuhi Sebagian Persyaratan
Untuk Memperoleh Gelar Ahli Madya**

Pada

Bidang Studi Komputer Kontrol

Program Studi D3 Teknik Elektro

Fakultas Teknologi Industri

Institut Teknologi Sepuluh Nopember

Menyetujui :

Dosen Pembimbing,



**SURABAYA
JANUARI, 2015**

PROTOTYPE SIMULASI SISTEM PENYALURAN AIR BERSIH DI DESA BERBASIS RFID

Nama Mahasiswa : Gamal Armando
NRP : 2211 030 066
Nama Mahasiswa : Muhammad Imam Ainul Hidayat
NRP : 2211 030 116
Dosen Pembimbing : Slamet Budiprayitno, ST., MT.
NRP : 19781113 201012 1 002

ABSTRAK

Air bersih di Indonesia menjadi masalah yang susah diselesaikan, terutama di desa – desa yang tertinggal. Penyaluran air bersih untuk desa-desa yang tertinggal dan jauh dari sumber mata air terkadang juga kurang merata. Dengan adanya permasalahan seperti ini diharapkan adanya pemerataan penyaluran air bersih di desa-desa yang tertinggal atau kesulitan air bersih, oleh karena itu kami membuat sebuah tugas akhir yang dapat memberi solusi terhadap permasalahan ini.

Tugas akhir ini membahas tentang prototipe simulasi sistem penyaluran air bersih di desa berbasis *Radio Frequency Identification* (RFID). Konsep penyaluran yang kami maksud adalah dengan diberlakukannya kartu sakti yang dilengkapi dengan *Radio Frequency Identification* (RFID). Perangkat elektronik ini kami namakan RFID-Air. Kartu Perangkat elektronik ini selain dipakai sebagai salah satu syarat untuk mendapatkan air bersih yang di kemas dalam jirigen. Manfaat yang dapat diambil dengan diterapkannya penggunaan RFID-Air adalah untuk memonitoring penyaluran air di setiap desa dan untuk memberikan kenyamanan dan kepastian setiap penduduk desa dapat memperoleh air bersih sesuai dengan yang telah dijatahkan untuk setiap penduduk (*individu*). Masyarakat dihadapkan pada sistem komputasi yang tegas dan tidak kompromi.

Kata Kunci : Air, Rfid, Database

{Halaman ini sengaja dikosongkan}

PROTOTYPE SIMULATION OF WATER DISTRIBUTION SYSTEM BASED ON VILLAGE WITH RFID BASED

Name : Gamal Armando
ID Number : 2211 030 066
Name : Muhammad Imam Ainul Hidayat
ID Number : 2211 030 116
Supervisor : Slamet Budiprayitno, ST., MT.
ID Number : 19781113 201012 1 002

ABSTRACT

Clean water in Indonesia has become a problem that is difficult to be resolved, particularly in the village - the village is left behind. The distribution of clean water to villages left behind and away from water sources sometimes also less prevalent. With the existence of such problems is expected that the equalization pernyaluran clean water in villages left behind or clean water shortages, therefore we make a final project that can provide a solution to this problem.

This final project discusses the prototype simulating the water distribution system in the village-based Radio Frequency Identification (RFID). The concept of distribution, we refer to the enactment of magic cards equipped with Radio Frequency Identification (RFID). These electronic devices we call RFID-Water. These electronic devices other than the card is used as one of the requirements to obtain clean water in containers in cisterns. The benefits that can be taken with the application of the use of RFID-Water is for monitoring water distribution in each village and to provide comfort and reassurance each resident can obtain clean water in accordance with the previously allocated for each population (people). Society faced with computing systems is firm and not compromise.

Keywords: *Air, Rfid, Database*

{Halaman ini sengaja dikosongkan}

KATA PENGANTAR

Segala puji dan syukur kehadiran Allah SWT atas segala rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan tugas akhir ini dengan baik dan lancar.

Dalam menyelesaikan buku tugas akhir ini penulis tidak lepas dari bimbingan, bantuan dan partisipasi dari berbagai pihak baik secara moril maupun materiil. Oleh karena itu penulis dalam kesempatan ini mengucapkan terima kasih kepada :

1. Bapak Slamet Budiprayitno, ST., MT. Selaku dosen pembimbing Tugas akhir atas dukungan motivasi dan semangat yang telah bapak berikan kepada kami.
2. Ayah, Ibu, Kakak, Adik bagi kedua penulis yang telah membantu baik dari segi materi, semangat maupun motivasinya.
3. Teman-teman seperjuangan penulis di D3 Teknik Elektro FTI ITS angkatan 2011 Winning'11 yang telah berjuang bersama-sama untuk wisudah dan menggapai mimpi bersama.
4. *HARDWORKS TECHNOLOGY* yang telah membantu kami dengan tulus.
5. Seluruh alumni D3 Teknik Elektro yang telah membantu dalam memotivasi dan memberikan dukungan untuk tugas akhir ini.
6. Partnerku (bagi keduanya) atas kekompakan dalam tim, perjuangan tanpa lelah, semangat, modal serta jerih payah yang telah kita lakukan demi mengerjakan tugas akhir ini.

Dengan segala kerendahan hati, kami berharap apa yang ada dalam buku tugas akhir ini dapat bermanfaat, dan berguna sebagai sumbangan pikiran bagi kita semua dalam berprestasi turut mengisi pembangunan bangsa dan negara.

Surabaya, Januari 2015

Penyusun

{Halaman ini sengaja dikosongkan}

DAFTAR ISI

| | |
|---|------|
| JUDUL | i |
| PENGESAHAN | iii |
| ABSTRAK | v |
| ABSTRACT | vii |
| KATA PENGANTAR | ix |
| DAFTAR ISI | xi |
| DAFTAR GAMBAR | xiii |
| DAFTAR TABEL | xv |
| | |
| BAB I PENDAHULUAN | 1 |
| 1.1 Latar Belakang | 1 |
| 1.2 Perumusan Masalah | 1 |
| 1.3 Batasan Masalah | 2 |
| 1.4 Tujuan | 2 |
| 1.5 Metodologi | 3 |
| 1.6 Sistematika Laporan | 3 |
| 1.7 Relevansi | 4 |
| | |
| BAB II TEORI PENUNJANG | 5 |
| 2.1 <i>Visual Basic</i> | 5 |
| 2.2 Dasar Mikrokontroler ATmega16 | 5 |
| 2.2.1 Arsitektur AVR ATmega16 | 5 |
| 2.2.2 Memori ATmega16 | 7 |
| 2.2.3 Pin-Pin ATmega16 | 8 |
| 2.2.4 Deskripsi Mikrokontroler ATmega16 | 9 |
| 2.3 RFID (<i>Radio Frequency Identification</i>) | 11 |
| 2.3.1 Beberapa Tipe RFID tag | 11 |
| 2.3.2 Sistem Kerja RFID | 12 |
| 2.3.3 Frekuensi Radio & Jangkauan | 15 |
| 2.3.4 Cara Kerja RFID | 16 |
| 2.4 LCD | 16 |
| | |
| BAB III PERANCANGAN DAN PEMBUATAN ALAT | 19 |
| 3.1 Perancangan Perangkat Mekanik | 20 |
| 3.2 Perancangan Perangkat Keras | 21 |
| 3.2.1 Sistem Minimum Mikrokontroler ATmega16 | 21 |
| 3.2.1.1 Sistem Minimum Mikrokontroler ATmega 16 Sebagai Penerima Data RFID. 21 | |
| 3.2.1.2 Sistem Minimum Mikrokontroler | |

| | |
|--|-----------|
| ATMega16 Pengaktifan Sistem RFID | 22 |
| 3.2.2 Rangkaian RFID RC 522 | 23 |
| 3.3. Perancangan Perangkat Lunak..... | 26 |
| 3.3.1 Perancangan Perangkat Lunak untuk RFID | 26 |
| 3.3.2 Perancangan Perangkat Lunak untuk Sistem | 27 |
| 3.3.2.1 Perancangan Perangkat Lunak RFID..... | 28 |
| 3.4 Perancangan Pembuatan <i>Hardware</i> | 29 |
| 3.5 Perancangan Pembuatan <i>Interface</i> | 31 |
| 3.5.1 Rancangan <i>Sign Up Page</i> | 31 |
| 3.5.2 Rancangan Menu Utama | 32 |
| 3.5.3 Rancangan Menu Koneksi | 32 |
| 3.5.4 Rancangan Registrasi | 33 |
| 3.5.4.1 Registrasi Pengguna RFID <i>Water</i> | 33 |
| 3.5.5 Rancangan <i>Record</i> Penduduk | 34 |
| 3.3.6 Rancangan Data Pengguna RFID <i>Water</i> | 34 |
| 3.5.7 Perancangan Sistem Penggunaan Kartu <i>tag</i> RFID Sebagai ID Pengguna RFID <i>Water</i> | 35 |
| 3.6 Cara Kerja Sistem..... | 35 |
| BAB IV PENGUJIAN DAN ANALISA DATA | 37 |
| 4.1. Desain Miniatur Prototipe Pedesaan..... | 37 |
| 4.2. Pengujian Rangkaian Mikrokontroler Atmega16 | 38 |
| 4.3. Pengujian pada RFID RC522 | 41 |
| 4.4. Pengujian Tampilan <i>Visual Studio</i> 2012 dan <i>Database</i> .. | 44 |
| 4.4.1. Tampilan Layar Implementasi Program..... | 44 |
| 4.5. Pengujian RFID <i>Water</i> | 47 |
| 4.6. Pengujian Sistem Keseluruhan | 48 |
| BAB V KESIMPULAN DAN SARAN | 53 |
| 5.1 Kesimpulan | 53 |
| 5.2 Saran | 53 |
| DAFTAR PUSTAKA | 55 |
| DAFTAR RIWAYAT HIDUP | |
| LAMPIRAN | |

DAFTAR TABEL

| | | |
|------------------|--|----|
| Tabel 2.1 | Pin Pada RFID-RC 522 | 13 |
| Tabel 3.1 | Deskripsi dan Format Data <i>Output</i> RC522 | 25 |
| Tabel 4.1 | Hasil Pengukuran Mikrokontroler pada <i>PORT A</i> | 39 |
| Tabel 4.2 | Hasil Pengukuran Mikrokontroler pada <i>PORT B</i> | 40 |
| Tabel 4.3 | Hasil Pengukuran Mikrokontroler pada <i>PORT C</i> | 40 |
| Tabel 4.4 | Hasil Pengukuran Mikrokontroler pada <i>PORT D</i> | 41 |
| Tabel 4.5 | Hasil Pengujian Jarak Pada Kemiringan 0° | 43 |
| Tabel 4.6 | Hasil Pengujian Jarak Pada Kemiringan 30° - 45° | 43 |

{Halaman ini sengaja dikosongkan}

DAFTAR GAMBAR

| | | |
|-------------|--|----|
| Gambar 2.1 | Diagram Blok ATmega16..... | 6 |
| Gambar 2.2 | Memori Program Mikrokontroler ATmega16 | 7 |
| Gambar 2.3 | Konfigurasi Pin Mikrokontroler ATmega16..... | 8 |
| Gambar 2.4 | Arsitektur CPU ATmega16 | 9 |
| Gambar 2.5 | RFID RC522 | 12 |
| Gambar 2.6 | <i>Pin Out</i> RFID | 14 |
| Gambar 2.7 | Skematik Prinsip Kerja RFID..... | 16 |
| Gambar 2.8 | LCD 2x16..... | 17 |
| Gambar 3.1 | Blok Fungsional Alat | 19 |
| Gambar 3.2 | Rancangan Sistem Mekanik | 20 |
| Gambar 3.3 | Sistem Minimum ATmega16 Sebagai Penerima Data RFID | 22 |
| Gambar 3.4 | Simulasi Proteus Kalkulator R-W | 23 |
| Gambar 3.5 | RFID <i>tag</i> | 24 |
| Gambar 3.6 | RC522 <i>Pin Out</i> | 24 |
| Gambar 3.7 | Skematik Rangkaian RFID | 25 |
| Gambar 3.8 | <i>Flow Chart</i> Sistem Perangkat Lunak | 27 |
| Gambar 3.9 | <i>Flow Chart</i> Sistem RFID dan Database | 28 |
| Gambar 3.10 | <i>Hardware</i> RFID <i>Water</i> | 29 |
| Gambar 3.11 | Perancangan Tempat Peletakan <i>Hardware</i> | 30 |
| Gambar 3.12 | Rancangan <i>Sign-Up Page</i> | 31 |
| Gambar 3.13 | Rancangan Menu Koneksi | 32 |
| Gambar 3.14 | <i>Form</i> Pendaftaran Pengguna RFID <i>Water</i> | 33 |
| Gambar 3.15 | Rancangan <i>Record</i> Pengguna RFID <i>Water</i> | 33 |
| Gambar 3.16 | Rancangan Data Pengguna RFID <i>Water</i> | 34 |
| Gambar 3.17 | Kondisi Saat Belum Mengisi Jirigen Air Bersih | 35 |
| Gambar 3.18 | Kondisi Setelah Mengisi Jirigen Air Bersih..... | 35 |
| Gambar 4.1 | Miniatur RFID <i>Water</i> | 37 |
| Gambar 4.2 | RFID <i>Water</i> Tampak Atas..... | 38 |
| Gambar 4.3 | Mengaktifkan <i>Port Out</i> pada Atmega16 | 38 |
| Gambar 4.4 | Aplikasi <i>Hyper Terminal</i> | 41 |
| Gambar 4.5 | <i>Connect to COM</i> | 42 |
| Gambar 4.6 | Tampilan Pada <i>Hyper Terminal</i> | 42 |
| Gambar 4.7 | Pengecekan RFID Jarak Kemiringan 0 | 42 |
| Gambar 4.8 | Pengecekan RFID Jarak Kemiringan 30 ⁰ – 45 ⁰ | 43 |
| Gambar 4.9 | <i>Database VB.NET</i> | 44 |
| Gambar 4.10 | Tampilan Layar <i>Login Form</i> | 44 |

| | | |
|-------------|---|----|
| Gambar 4.11 | Tampilan Utama | 45 |
| Gambar 4.12 | Tampilan Koneksi..... | 45 |
| Gambar 4.13 | <i>Log-in Form</i> Pengisian COM | 46 |
| Gambar 4.14 | Tampilan Regristasi Pengguna RFID Water | 46 |
| Gambar 4.15 | Tampilan Data Setelah Terdaftar | 47 |
| Gambar 4.16 | Tampilan Awal RFID <i>Water</i> | 47 |
| Gambar 4.17 | Pilihan Pembelian Jirigen / Rupiah..... | 48 |
| Gambar 4.18 | Perbedaan 2 Kondisi | 48 |
| Gambar 4.19 | Pilih Jumlah Jirigen Yang Diinginkan | 49 |
| Gambar 4.20 | Nilai Rupiah Pembelian | 50 |
| Gambar 4.21 | <i>Counter</i> Keluaran Air | 50 |
| Gambar 4.22 | Status Hasil Sisa Penyaluran Air | 51 |

BAB I

PENDAHULUAN

1.1 Latar Belakang

Air merupakan kebutuhan kehidupan yang sangat dibutuhkan, sehingga perlu adanya perhatian khusus terkait dengan air, didesa sedikit terdapat air bersih dan perlu adanya penyaluran air bersih. Dengan adanya penyaluran air bersih warga desa akan lebih terpenuhi kebutuhannya karena air bersih dapat dipakai bermacam-macam manfaat yaitu untuk mandi, masak, cuci, dll, selain itu kebersihan air juga dapat meningkatkan kesehatan masyarakat desa yang sebelumnya menggunakan air biasa yang bisa menjadi penyebab penyakit dan sekarang dengan adanya penyaluran air bersih dapat meningkatkan kualitas kesehatan masyarakat.

Di desa tidak semua orang kurang mampu ada juga orang yang mampu sehingga orang mampu ini memilih air bersih untuk dijadikan kebutuhannya sehari hari, akan tetapi orang-orang yang kurang mampu akan memakai air apa adanya bisa dari sumur, sungai dll, sehingga belum bisa dijamin kesegaran dan kesehatan dari air tersebut. Perlu adanya penyaluran dengan sistem yang bagus untuk pemeratakan penduduk desa agar mendapatkan air bersih.

Hal ini membuat perlunya perhatian dari masyarakat untuk mengadakan sistem penyaluran air bersih, pada praktiknya biasanya terjadi ketidak-merataan dalam penyaluran untuk itu perlu diadakan simulasi sistem penyaluran air berbasis RFID yang berguna untuk mendata penduduk serta dapat digunakan untuk mengambil air bersih bagi penduduk yang layak mendapatkan air bersih dan orang-orang yang mampu untuk membeli air bersih akan dikenakan biaya pembelian air bersih yang sudah di monitoring melalui *database*. Kartu yang di gunakan juga berfungsi untuk data penduduk yang kami beri nama R-W (RFID *Water*) *Card*.

1.2 Perumusan Masalah

Beberapa masalah yang akan diselesaikan pada proses implementasi rancangan RFID *Water* menggunakan RFID RC 522 yang di hubungkan mikrokontroler AVR yang terintegrasi dengan *database* antara lain :

1. Hasil pengujian sistem pada RFID dengan mikrokontroler AVR yang berubah setiap saat.

2. Algoritma pengiriman data dari RFID ke mikrokontroler memerlukan ketelitian.
3. Hasil pengujian penerimaan data mikrokontroler dari *software* matlab diperoleh dengan baik.
4. Hasil pengujian akurasi kemiringan pada *tag* RFID di perlukan ketelitian.

1.3 Batasan Masalah

Dalam perancangan ini perlu diberikan beberapa batasan permasalahan dengan tujuan agar pembahasan tidak meluas dan menyimpang dari tujuan. Adapun batasan masalah dari alat yang dirancang pada tugas akhir ini yaitu:

1. Software yang digunakan untuk RFID adalah AVR.
2. Menggunakan rumus yang sederhana untuk menjalankan *counter*.
3. Menggunakan rangkaian *relay* untuk menjalankan pompa airnya.
4. Miniatur desa untuk dibuat agar *prototype* mirip dengan penggunaan aslinya pada saat pengisian air bersih.
5. Pada pengerjaan tugas akhir ini tidak menggunakan pompa yang sesuai dengan pompa aslinya sehingga tidak dapat mengeluarkan debit air sesuai liter pada display LED.
6. Menggunakan komunikasi serial untuk menghubungkan RFID *Water* ke *database*.
7. RFID yang digunakan memiliki frekuensi yang terbilang rendah.
8. Akses *database* hanya untuk konfirmasi sn pada kartu RFID.

1.4 Tujuan

Tujuan yang hendak dicapai melalui pembuatan tugas akhir ini adalah sebagai berikut :

1. Membuat perancangan alat untuk penyaluran air bersih di desa secara tepat sesuai kebutuhan.
2. Memanfaatkan keahlian dibidang elektronika untuk membantu penduduk desa mendapatkan air bersih sesuai kebutuhan.
3. Melestarikan Budaya taat peraturan desa dengan kartu R-W yang menanamkan nilai disiplin di dalamnya.

1.5 Metodologi

Dalam pelaksanaan tugas akhir ini ada beberapa kegiatan yang dapat diuraikan sebagai berikut :

1.Studi Pustaka dan Survei Data:

Kegiatan ini berupa mencari informasi baik dari literatur di perpustakaan, internet ataupun survey lapangan tentang sistem yang akan digunakan pada perancangan pembuatan tugas akhir ini. Penggalan informasi berupa bentuk areal fisik, mekanisme kerja, metode – metode pengolahan citra digital dan perangkat elektronik yang digunakan.

2.Perancangan dan Pembuatan Alat:

Perancangan disusun atas beberapa bagian elektronik serta dasar mekanik yang mirip dengan kondisi ideal. Bentuk fisik dan pemilihan bahan dipertimbangkan agar dapat menghasilkan pergerakan yang presisi.

3.Perancangan dan Pembuatan *Software*:

Perangkat lunak yang dimaksud, yaitu perangkat lunak berupa program-program. Ini dimaksudkan agar alat yang dibuat bisa berjalan sesuai dengan apa yang diinginkan dalam tugas akhir ini.

4.Uji Coba dan Analisis Data:

Pengujian dilakukan di Laboratorium Elka Dasar BB102 secara bertahap baik pada sisi *software* RFID, elektronik dan mekanik, melakukan analisa untuk menghasilkan hasil yang tepat.

5.Penyusunan Laporan:

Dilakukan setelah seluruh proses pembuatan alat dan penelitian selesai. Laporan dibuat berdasarkan kenyataan yang telah dilakukan dan bisa dipertanggung-jawabkan kebenarannya.

1.6 Sistematika Laporan

Bab I : PENDAHULUAN

Dalam bab ini diuraikan tentang latar belakang, permasalahan, batasan masalah, maksud dan tujuan, metodologi, sistematika laporan, serta relevansi penulisan pada tugas akhir ini.

Bab II : TEORI PENUNJANG

Dalam bab ini akan ini dijelaskan mengenai konsep-konsep yang mendasari perancangan pada tugas akhir ini, yang meliputi dasar pengolahan citra digital beserta penjelasan jenis pengolahan citra digital yang digunakan dalam penelitian ini, serta teori – teori dari mikrokontroler ATmega16, komunikasi serial RS 232, RFID (*Radio Frequency Identification*) dan LCD.

Bab III : PEMBUATAN ALAT

Dalam bab ini dijelaskan mengenai perancangan sistem yang meliputi perancangan perangkat lunak dan perangkat keras. Perangkat lunak meliputi pemrograman RFID dan algoritma penerimaan data mikrokontroler, perangkat keras meliputi rangkaian-rangkaian dan mekanik.

Bab IV: PENGUKURAN DAN ANALISA

Dalam bab ini membahas tentang pengujian pada masing-masing sub sistem dan sistem secara keseluruhan sehingga didapatkan data-data dan analisa.

Bab V: PENUTUP

Dalam bab ini menjelaskan kesimpulan dari hasil pembuatan alat dan saran-saran untuk pengembangan alat ini lebih lanjut.

1.7 Relevansi

Manfaat yang didapat dalam pengerjaan tugas akhir ini bagi kami sendiri adalah mengasah kemampuan kami dalam bidang elektro. Dengan membangun sebuah sistem simulasi RFID *Water* ini, kami dapat mengimplementasikan berbagai ilmu yang kami dapat semasa kuliah. Beberapa diantaranya adalah pembacaan RFID, mikrokontroler, dan komunikasi data. Selain itu, sistem yang kami buat ini diharapkan dapat bekerja sebagai sarana untuk memonitoring penyaluran air bersih di desa.

BAB II

TEORI PENUNJANG

2.1 Visual Basic ^[1]

Microsoft Visual Basic (sering disingkat sebagai VB) merupakan sebuah bahasa pemrograman yang menawarkan *Integrated Development Environment* (IDE) visual untuk membuat program perangkat lunak berbasis sistem operasi *Microsoft Windows* dengan menggunakan model pemrograman (COM), *Visual Basic* merupakan turunan bahasa pemrograman BASIC dan menawarkan pengembangan perangkat lunak komputer berbasis grafik dengan cepat, Beberapa bahasa skrip seperti *Visual Basic for Applications* (VBA) dan *Visual Basic Scripting Edition* (VBScript), mirip seperti halnya *Visual Basic*, tetapi cara kerjanya yang berbeda. Para *programmer* dapat membangun aplikasi dengan menggunakan komponen-komponen yang disediakan oleh *Microsoft Visual Basic*. Program-program yang ditulis dengan *Visual Basic* juga dapat menggunakan Windows API, tapi membutuhkan deklarasi fungsi luar tambahan. Dalam pemrograman untuk bisnis, *Visual Basic* memiliki pangsa pasar yang sangat luas. Dalam sebuah survey yang dilakukan pada tahun 2005, 62% pengembang perangkat lunak dilaporkan menggunakan berbagai bentuk *Visual Basic*, yang diikuti oleh C++, JavaScript, C#, dan Java.

Beberapa kemampuan atau manfaat dari *Visual Basic* diantaranya seperti:

1. Untuk membuat program aplikasi berbasis windows.
2. Untuk membuat objek-objek pembantu program seperti misalnya *control ActiveX*, *file Help*, aplikasi internet, dan sebagainya.
3. Menguji program (*debugging*) dan menghasilkan program akhir berakhiran EXE yang bersifat *executable*, atau dapat langsung dijalankan.

2.2 Dasar Mikrokontroler ATMega16^[2]

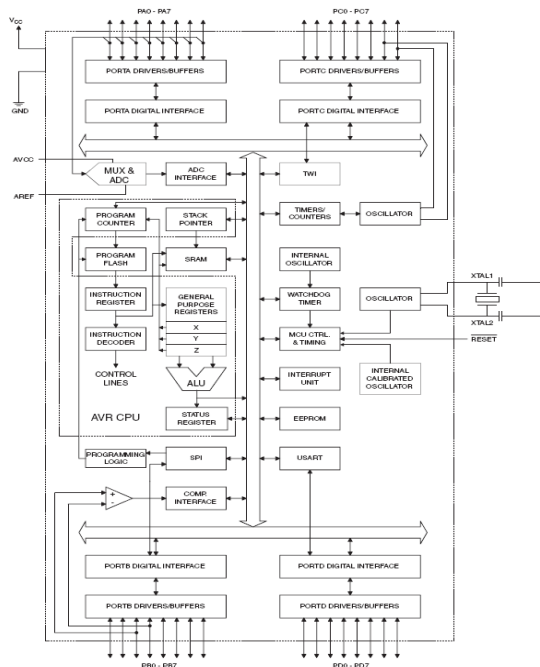
Dasar Mikrokontroler ATMega16 dibahas menjadi 3 yaitu Arsitektur AVR ATMega16, Pin-Pin ATMega16 dan Deskripsi Mikrokontroler ATMega16.

2.2.1 Arsitektur AVR ATMega16

AVR merupakan seri mikrokontroler CMOS 8-bit buatan ATmel, berbasis arsitektur RISC (*Reduced Instruction Set Computer*). Hampir

semua instruksi dieksekusi dalam satu siklus *clock*. AVR mempunyai 32 *register general-purpose*, *timer/counter fleksibel* dengan *mode compare*, *interrupt internal* dan *external*, serial UART, programmable *Watchdog Timer*, dan *mode power saving*. Beberapa diantaranya mempunyai ADC dan PWM *internal*. AVR juga mempunyai *In-System Programmable Flash on-chip* yang memungkinkan memori program untuk diprogram ulang dalam sistem menggunakan hubungan serial SPI. *Chip AVR* yang digunakan untuk tugas akhir ini adalah ATmega16.

ATmega16 adalah mikrokontroler CMOS 8-bit daya-rendah berbasis arsitektur RISC yang ditingkatkan. Kebanyakan instruksi dikerjakan pada satu siklus *clock*, ATmega16 mempunyai *throughput* mendekati 1 MIPS per MHz membuat disainer sistem untuk mengoptimasi konsumsi daya *versus* kecepatan proses. Diagram blok dari mikrokontroler dapat dilihat pada Gambar 2.1.



Gambar 2.1 Diagram Blok ATmega16

2.2.2 Memori ATmega16

Mikontroler ATmega16 memiliki ruang pengalamatan memori data dan memori program yang terpisah. Sebagai tambahan, ATmega16 memiliki *fitur* suatu Memori *EEPROM* untuk penyimpanan data. Sehingga ATmega16 memiliki tiga jenis memori yaitu memori program, memori data, dan memori *EEPROM* dan ketiganya memiliki ruang sendiri dan terpisah.

1. Memori Program

Mikontroler ATmega16 memiliki kapasitas memori program sebesar 8 *Kbyte* yang terpetakan dari alamat 0000h-0FFFh dimana masing-masing alamat memiliki lebar data 16-bit. Sehingga organisasi memori program seperti ini sering ditulis dengan 4K x 16 bit.

Memori program terbagi menjadi dua bagian program yaitu bagian program *boot* dan bagian program aplikasi. Jika kita tidak menggunakan *fitur boot loader flash* maka semua kapasitas memori program di atas dapat digunakan untuk program aplikasi. Tetapi jika *fitur boot loader flash* digunakan maka terjasdi pembagian ukuran kedua bagian ini yang ditentukan oleh *BOOTSZ fuse*. Flash memori mempunyai suatu daya tahan sedikitnya 10.000 *write/erase Cycles*. Untuk memori program Mikrokontroler dapat dilihat pada Gambar 2.2:



Gambar 2.2 Memori Program Mikrokontroler ATmega16

2. Memori Data

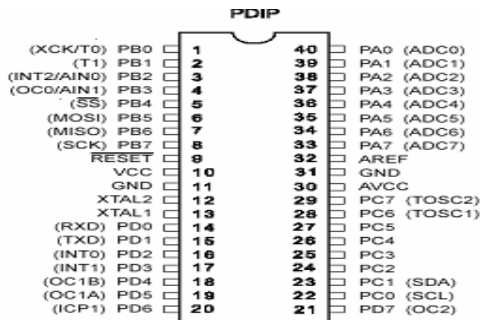
Mikrokontroler ATmega16 memiliki kapasitas memori data sebesar 608 *Byte*. Memori data terbagi menjadi 3 bagian, yaitu 32 *register* umum, 64 buah *register* I/O, dan 512 *byte SRAM Internal*. *Register* umum menempati *space* data pada alamat terbawah, yaitu 0000h sampai 001Fh. Untuk *register* khusus yang menangani I/O dan kontrol terhadap mikrokontroler menempati 64 alamat berikutnya, yaitu mulai dari 0020h hingga 005Fh. *Register* tersebut merupakan *register* yang khusus digunakan untuk mengatur fungsi terhadap berbagai *peripheral* mikrokontroler, seperti kontrol *register*, *timer/counter*, fungsi – fungsi I/O, *ADC*, *SPI* dan sebagainya. Alamat memori berikutnya digunakan untuk *SRAM* 512 *byte*, yaitu pada lokasi 0060h sampai 025Fh.

3. Memori EEPROM

Mikrokontroler ATmega8535 memiliki memori *EEPROM* sebesar 512 *Byte* yang terpisah dari memori program dan memori data. Memori *EEPROM* ini hanya dapat diakses dengan menggunakan *register-register* I/O yaitu *register EEPROM address (EEARH-EEARL)*, *register EEPROM data (EEDR)*, dan *register EEPROM control (EECR)*.

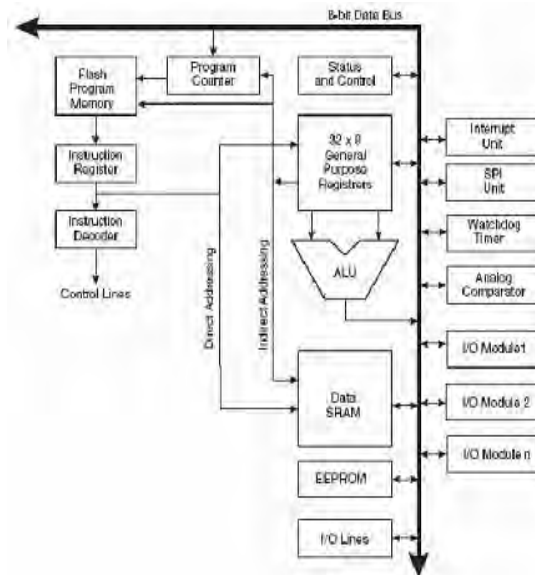
2.2.3 Pin-Pin ATmega16

Konfigurasi Pin Mikrokontroler ATmega16 dengan kemasan 40-pin DIP (*dual in-line package*) dapat dilihat pada Gambar 2.3:



Gambar 2.3 Konfigurasi Pin Mikrokontroler ATmega16

Untuk memaksimalkan performa, AVR menggunakan arsitektur *Harvard* (dengan memori dan bus terpisah untuk program dan data). Arsitektur CPU dari AVR ditunjukkan oleh Gambar 2.4 Instruksi pada memori program dieksekusi dengan *pipelining single level*. Selagi sebuah instruksi sedang dikerjakan, instruksi berikutnya diambil dari memori program.



Gambar 2.4 Arsitektur CPU ATmega16

2.2.4 Deskripsi Mikrokontroler ATmega16

Mikrokontroler ATmega16 terdiri dari 40 pin dan 4 buah *Port*, yaitu *Port A*, *Port B*, *Port C* dan *Port D*. Keempat *Port* tersebut memiliki fungsi sebagai masukan atau keluaran dari mikrokontroler dan juga memiliki fungsi-fungsi khusus yang dapat dijelaskan dibawah ini :

1. VCC (*power supply*).
2. GND (*ground*).
3. *Port A* (PA7..PA0).
4. *Port A* berfungsi sebagai masukan analog pada A/D Konverter. *Port A* juga berfungsi sebagai suatu *Port I/O* 8-bit dua arah, jika A/D Konverter tidak digunakan. Pin - pin *Port*

dapat menyediakan resistor *internal pull-up* (yang dipilih untuk masing-masing bit). *Port A* keluaran *buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Ketika pin PA0 ke PA7 digunakan sebagai masukan dan secara *external* ditarik rendah, pin – pin akan memungkinkan arus sumber jika resistor *internal pull-up* diaktifkan. Pin *Port A* adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.

5. *Port B* adalah suatu *Port I/O* 8-bit dua arah dengan resistor *intenal pull-up* (yang dipilih untuk beberapa bit). *Port B* keluaran *buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai masukan, pin *Port B* yang secara *external* ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. pin *Port B* adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.
6. *Port C* adalah suatu *Port I/O* 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). *Port C* keluaran *buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai masukan, pin *Port C* yang secara *external* ditarik rendah akan arus sumber jika resistor *pull-up* diaktifkan. Pin *Port C* adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.
7. *Port D* adalah suatu *Port I/O* 8-bit dua arah dengan resistor *internal pull-up* (yang dipilih untuk beberapa bit). *Port D* keluaran *buffer* mempunyai karakteristik gerakan simetris dengan keduanya *sink* tinggi dan kemampuan sumber. Sebagai masukan, pin *Port D* yang secara *external* ditarik rendah akan arus sumber resistor *pull-up* diaktifkan. Pin *Port D* adalah *tri-stated* manakala suatu kondisi *reset* menjadi aktif, sekalipun waktu habis.
8. *RESET* (*Reset* masukan).
9. XTAL1 (Masukan *Oscillator*).
10. XTAL2 (Keluaran *Oscillator*).
11. AVCC adalah pin penyedia tegangan untuk *Port A* dan A/D Converter.
12. AREF adalah pin *referensi analog* untuk A/D konverter.

2.3 RFID (*Radio Frequency Identification*)^[3]

RFID atau *Radio Frequency Identification*, adalah suatu metode yang mana bisa digunakan untuk menyimpan atau menerima data secara jarak jauh dengan menggunakan suatu piranti yang bernama *RFID tag* atau *transponder*. Suatu *RFID tag* adalah sebuah benda kecil, misalnya berupa stiker adesif, dan dapat ditempelkan pada suatu barang atau produk. *RFID tag* berisi antena yang memungkinkan mereka untuk menerima dan merespon terhadap suatu *query* yang dipancarkan oleh suatu *RFID transceiver*.

2.3.1 Beberapa Tipe RFID Tag

RFID *tag* dapat bersifat aktif atau pasif. RFID *tag* yang pasif tidak memiliki *Power Supply* sendiri. Dengan hanya berbekal induksi listrik yang ada pada antena yang disebabkan oleh adanya frekuensi radio scanning yang masuk, sudah cukup untuk memberi kekuatan yang cukup bagi RFID *tag* untuk mengirimkan respon balik. Sehubungan dengan power dan biaya, maka respon dari suatu RFID yang pasif biasanya sederhanya, hanya nomor ID saja. Dengan tidak adanya *Power Supply* pada RFID *tag* yang pasif maka akan menyebabkan semakin kecilnya ukuran dari RFID *tag* yang mungkin dibuat. Beberapa RFID komersial yang saat ini sudah beredar di pasaran ada yang bisa diletakkan di bawah kulit. Pada tahun 2005 tercatat bahwa RFID *tag* terkecil berukuran 0.4 mm x 0.4 mm dan lebih tipis daripada selembar kertas. Dengan ukuran sekian maka secara praktis benda tersebut tidak akan terlihat oleh mata. RFID *tag* yang pasif ini memiliki jarak jangkauan yang berbeda mulai dari 10 mm sampai dengan 6 meter. RFID *tag* yang aktif, di sisi lain harus memiliki *Power Supply* sendiri dan memiliki jarak jangkauan yang lebih jauh. Memori yang dimilikinya juga lebih besar sehingga bisa menampung berbagai macam informasi di dalamnya. Sampai tulisan ini dipublikasikan, ukuran terkecil dari RFID *tag* yang aktif ini ada yang sebesar koin. Jarak jangkauan dari RFID *tag* yang aktif ini bisa sampai sekitar 10 meter dan dengan umur baterai yang bisa mencapai beberapa tahun lamanya. RFID *tag* yang pasif harganya bisa lebih murah untuk diproduksi dan tidak bergantung pada baterai. RFID *tag* yang banyak beredar sekarang adalah RFID *tag* yang sifatnya pasif. Ada empat macam RFID *tag* yang sering digunakan bila dikategorikan berdasarkan frekuensi radio :

1. *low frequency tag* (antara 125 ke 134 kHz).
2. *high frequency tag* (13,56 MHz).

3. UHF *tag* (868 sampai 956 MHz) *Microwave tag* (2,45 GHz).

UHF *tag* tidak bisa digunakan secara global, karena tidak ada peraturan global yang mengatur penggunaannya.

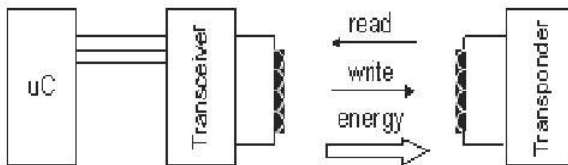
2.3.2 Sistem Kerja RFID (*Radio Frequency Identification*)

Suatu sistem RFID dapat terdiri dari beberapa komponen, seperti *tag*, *tag reader*, *tag programming station*, *circulation reader*, *sorting equipment* dan tingkat *inventory tag*. Keamanan dapat dicapai dengan dua cara. Pintu *security* dapat melakukan *query* untuk menentukan status keamanan atau RFID *tag* berisi *bit security* yang bisa menjadi *on* atau *off* pada saat didekatkan ke *reader station*.

Kegunaan dari sistem RFID ini adalah untuk mengirimkan data dari piranti *portabel*, yang dinamakan *tag*, dan kemudian dibaca oleh RFID *reader* dan kemudian diproses oleh aplikasi komputer yang membutuhkannya. Data yang dipancarkan dan dikirimkan tadi bisa berisi beragam informasi, seperti ID, informasi lokasi atau informasi lainnya seperti harga, warna, tanggal pembelian dan lain sebagainya. Seiring dengan perkembangan teknologi, maka teknologi RFID sendiripun juga berkembang sehingga nantinya penggunaan RFID bisa digunakan untuk kehidupan sehari-hari.

Sistem RFID sendiri umumnya terdiri dari dua bagian besar component, yaitu:

1. *Transponders* yang ditempelkan pada (yang akan diberi label).
2. *Readers* (biasa dikenal sebagai sensor RFID), yang digunakan untuk membaca identitas dari *transponders*. Seperti pada Gambar 2.5 dapat dilihat RFID RC522.



Gambar 2.5 RFID RC 522

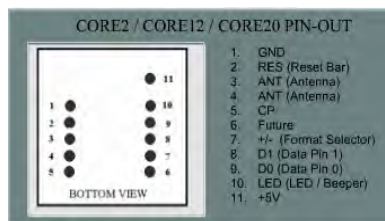
Suatu sistem RFID secara utuh terdiri atas 3 komponen yaitu :

1. *Tag* RFID, dapat berupa stiker, kertas atau plastik dengan beragam ukuran. Didalam setiap *tag* ini terdapat *chip* yang mampu menyimpan sejumlah informasi tertentu.

2. Terminal *Reader* RFID, terdiri atas RFID-*reader* dan antena yang akan mempengaruhi jarak optimal identifikasi. Terminal RFID akan membaca atau mengubah informasi yang tersimpan didalam *tag* melalui frekuensi radio.
3. *Host* Komputer, sistem komputer yang mengatur alur informasi dari item-item yang terdeteksi dalam lingkup sistem RFID dan mengatur komunikasi antara *tag* dan *reader*. Untuk lebih jelasnya dapat dilihat pada Tabel 2.1:

Tabel 2.1 Pin Pada RFID RC522

| No. Pin | Deskripsi | ASCII | Magnet Emulation | Wiegand 26 |
|---------|--|---------------------|------------------|------------------|
| Pin 1 | Zero Volt dan tuning kapasitor | GND 0V | GND 0V | GND 0V |
| Pin 2 | Strap to +5v | Reset bar | Reset bar | Reset bar |
| Pin 3 | Untuk Antena Eksternal dan Tuning Kapasitor Antena | Antena | Antena | Antena |
| Pin 4 | Untuk Antena | Antena | Antena | Antena |
| Pin 5 | Card present | Tidak | Ada | Tidak |
| Pin 7 | Format Selector (+/-) | Strap to GND | Strap to Pin 10 | Strap to Pin +5v |
| Pin 8 | Data 1 | CMOS | Clock | 1 output |
| Pin 9 | Data 0 | TTL Data (inverted) | Data | 0 output |
| Pin 10 | 3,1 kHz Logic | Beeper / LED | Beeper / LED | Beeper / LED |
| Pin 11 | DC Voltage Supply | +5V | +5V | +5V |



Gambar 2.6 Pin Out RFID

Pada Gambar 2.6 dapat dilihat *Pin Out* dari RFID. Label RFID atau yang biasa disebut RFID *tag* sendiri yang merupakan suatu *microchip* ber-antena dan juga disertakan pada satu unit barang. Sebuah *tag* RFID

memiliki *Electronic Product Code* (EPC) yang berisi nomor seri dan juga memiliki digit ekstra untuk mengidentifikasi suatu barang. RFID *tag* dapat bersifat aktif atau pasif, seperti dijelaskan dibawah ini:

1. RFID Pasif

RFID *tag* yang pasif tidak memiliki *power supply* maupun *transmitter* sendiri. *Transponder* RFID terdiri dari *microchip* yang menempel pada antenna bisa saja dibungkus dalam berbagai macam bentuk. Bentuk pembungkus yang digunakan tergantung pada jenis karakteristik aplikasi yang menggunakan RFID. Kartu RFID pasif dapat menggunakan *low frequency* (13,5MHz) atau UHF (860MHz-960MHz). Jenis frekuensi yang digunakan juga sangat bergantung pada karakteristik aplikasi karena tiap rentang frekuensi mempunyai karakteristik tertentu. Pada rentang frekuensi tertentu gelombang radio tidak dapat menembus benda logam atau air, rentang frekuensi juga mempunyai karakteristik jarak maksimum pancaran gelombang radio yang berbeda-beda. Kartu RFID pasif yang menggunakan UHF berharga lebih murah dan jangkauannya lebih luas (jangkauannya sampai dengan 3,33 meter). Untuk kartu RFID yang menggunakan *low frequency* hanya dapat dibaca pada jarak maksimal 0,3 meter dari piranti pembaca, sedang untuk *high frequency* dapat dibaca pada jarak 1 meter. Metode pengiriman data kartu RFID *tag* yang pasif ke piranti pembaca dibagi menjadi dua, yaitu :

- *Inductive Coupling*

Piranti pembaca yang membangkitkan elektromagnetik yang ada pada kartu RFID, dimana hasil induksi menjadi sumber tenaga bagi kartu RFID dengan piranti pembaca yang juga harus pendek agar induksi dapat ditangkap. *Inductive Coupling* ini digunakan pada kartu RFID dengan *low frequency* dan *high frequency*.

- *Propagation Coupling*

Energi yang digunakan berasal dari energi elektromagnetik (gelombang radio) yang dipancarkan oleh piranti pembaca. Kartu RFID kemudian akan mengumpulkan energi elektromagnetik ini untuk digunakan sebagai sumber daya mengirimkan data yang dimilikinya ke piranti pembaca. Mekanisme ini disebut dengan *backscatter*.

2. RFID Aktif

RFID *tag* yang aktif harus memiliki *power supply* sendiri dan memiliki jarak jangkauan yang lebih jauh. Memori yang dimiliki lebih besar sehingga dapat menampung berbagai macam informasi di

dalamnya. Jarak jangkauan dari RFID *tag* yang aktif bisa sampai sekitar 100 m.

2.3.3 Frekuensi Radio dan Jangkauan

Ada empat macam RFID *tag* yang sering digunakan berdasarkan frekuensi radio, yaitu :

1. Low Frequency RFID *tag* (125 – 134KHz)
Digunakan untuk identifikasi binatang, *beer keg tracking*, *keylock* pada mobil, dan sistem anti pencuri.
2. High Frequency RFID *tag* (13,56MHz)
Digunakan pada perpustakaan atau toko buku, *pallet tracking*, akses kontrol gedung, *apparel item tracking*, serta pengidentifikasi lencana.
3. UHF RFID *tag* (868 sampai 956 MHz)
Sering digunakan pada *pallet* dan pelacakan kontainer, pelacakan truk, trailer pada pelabuhan kapal laut.
4. Microwave RFID *tag* (2,45 GHz)
Sering digunakan dalam akses kontrol jarak jauh kendaraan bermotor.

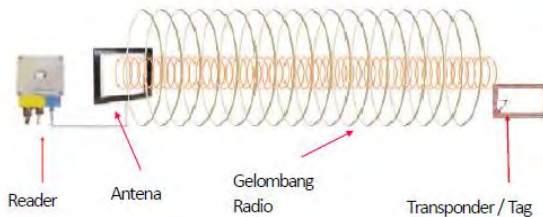
Jarak antara antenna pembaca RFID dengan *tag* secara langsung dipengaruhi oleh frekuensi yang digunakan. Frekuensi yang berbeda akan menghasilkan jangkauan yang berbeda pula. Ukuran antenna yang digunakan untuk transmisi data bergantung dari panjang gelombang elektromagnetik. Untuk frekuensi yang rendah, maka antenna harus dibuat dengan ukuran yang lebih besar dibandingkan dengan RFID dengan frekuensi tinggi.

2.3.4 Cara Kerja RFID

Sistem RFID dapat mengirimkan data dari sebuah alat yang dinamakan *tag* dan dibaca oleh RFID *reader* dan data yang didapat diproses oleh computer. Data yang dipancarkan dan dikirimkan tadi berisi beragam informasi, entah dari informasi lokasi dan sebagainya. Dimana RFID tersebut dapat mengirimkan data yang tersimpan didalam kartu atau *chip* yang dapat diaplikasikan untuk segala macam fungsi dan kegunaan

Prinsip kerja dari RFID adalah *reader* mengirimkan gelombang elektromagnetik ke segala arah dengan *range* dan frekuensi tertentu. Apabila suatu *tag* RFID dalam jangkauan *range* tersebut, maka akan menerima gelombang elektromagnetiknya dan akan mengirimkan

gelombang tersebut ke segala arah yang berisi informasi yang kemudian disimpan dalam *tag* dan biasanya berupa kode. Gelombang elektromagnetik akan diterima oleh *reader*, dimana *reader* menerima gelombang elektromagnetik balik yang juga berisi informasi. Informasi tersebut kemudian ditransmisikan ke computer melalui *port serial* ataupun mikrokontroler melalui USART. Pada mikrokontroler, informasi tersebut akan diolah. Skematik Prinsip kerja RFID dapat dilihat pada Gambar 2.7:



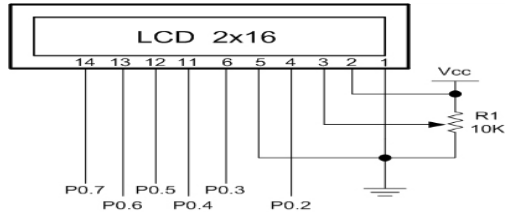
Gambar 2.7 Skematik Prinsip Kerja RFID

2.4 LCD ^[4]

Tampilan Kristal Cair (bahasa Inggris: *Liquid Crystal Display*) juga dikenal sebagai LCD adalah suatu jenis media tampilan yang menggunakan kristal cair sebagai penampil utama.

LCD yang digunakan pada tugas akhir ini adalah jenis LCD M1632. M1632 merupakan modul LCD dengan tampilan 16 x 2 baris dengan konsumsi daya rendah. Modul tersebut dilengkapi dengan mikrokontroler yang didesain khusus untuk mengendalikan LCD. Mikrokontroler HD44780 buatan Hitachi yang berfungsi sebagai pengendali LCD memiliki CGROM (*Character Generator Read Only Memory*), CGRAM (*Character Generator Random Access Memory*), dan DDRAM (*Display Data Random Access Memory*).

Pengiriman data ke LCD ada dua macam yaitu sebagai intruksi dan sebagai data character yang akan ditampilkan. keduanya dibedakan oleh sebuah kaki yang diberi nama RS(*Register Select*)dimana bila logika='1' (*high*) maka data yang diterima LCD adalah data *character* sedangkan bila RS='0' (*Low*) maka data yang diterima LCD adalah data intruksi. Intruksi diperlukan untuk inialisasi LCD, untuk meletakkan *cursor* pada baris dan kolom tertentu, untuk menghapus layar dll. Contoh LCD dapat dilihat pada Gambar 2.8.



Gambar 2.8 LCD 2x16

{Halaman ini sengaja dikosongkan}

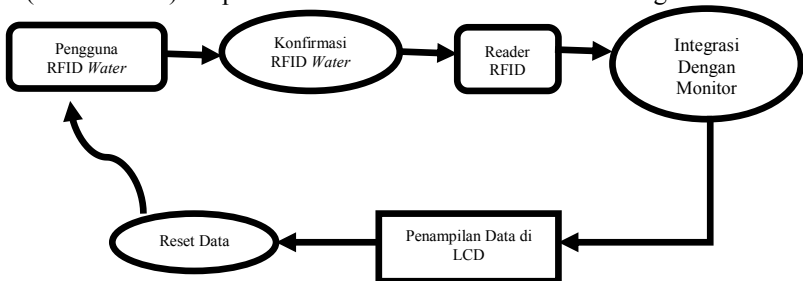
BAB III

PERANCANGAN DAN PEMBUATAN ALAT

Pada bab ini akan dibahas mengenai prototipe simulasi sistem penyaluran air bersih di desa berbasis RFID (*Radio Frequency Identification*) Menggunakan Mikrokontroler Atmega16 ter-integrasi pada *database*, baik perancangan perangkat keras (*hardware*), perancangan dan pembuatan perangkat lunak (*software*) serta perancangan mekanik yaitu pada pembuatan miniatur Penyaluran Air Bersih. Untuk perangkat keras meliputi :

1. Perancangan rangkaian *supply* menggunakan Travo 3Ampere dengan keluaran +5Volt.
2. Perancangan rangkaian sistem minimum ATmega16.
3. Perancangan rangkaian RFID RC 522.

Sedangkan untuk perancangan perangkat lunak (*software*) yang digunakan yaitu dengan menggunakan program *Codevision AVR* yang akan dimasukkan ke mikrokontroler yang digunakan untuk menampung data *tag* dari RFID RC 522 yang akan di kirim ke *visual basic* untuk di seleksi apakah data *tag* RFID ada atau tidak, apabila data *tag* RFID terdapat pada *visualbasic* maka *visualbasic* akan mengirim perintah untuk ke mikrokontroler lainnya untuk menjalankan sistem R-W (RFID-Water). Dapat dilihat di Gambar 3.1 untuk blok fungsional alat.



Gambar 3.1 Blok Fungsional Alat

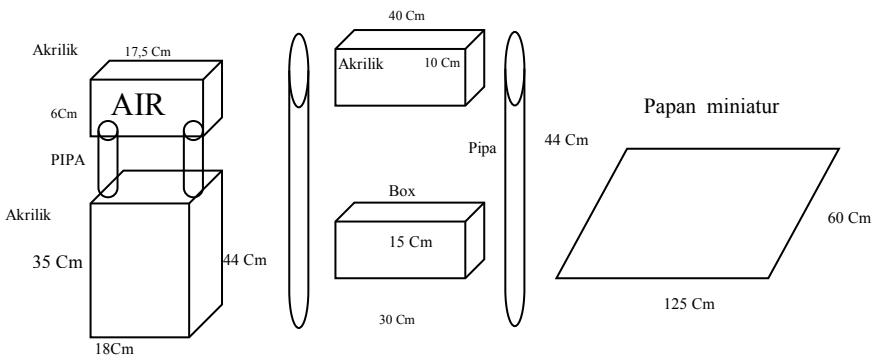
Alat ini dibuat untuk memonitoring penyaluran air bersih di desa setiap harinya dan berfungsi untuk memberi pengetahuan taat aturan RW (*RFID Water*) dengan adanya penyaluran air jenis air bersih menggunakan R-W. Konsep dasar alat ini adalah melakukan

monitoring terhadap penyaluran air bersih di desa dengan memanfaatkan adanya RFID yang diletakkan di kartu R-W. Penggunaan R-W akan diberi sebuah RFID *tag* yang kodenya telah dimasukkan dalam *database*, ketika pengguna mendekatkan RFID *tag* tersebut ke *reader* RFID maka mikrokontroler pertama akan langsung mengirim kode ke PC untuk dicatat penerima air bersih di desa. Ketika kode RFID *tag* masuk ke PC dan di *database* PC terdapat tag tersebut, maka penerima dapat menerima air bersih sesuai kebutuhan.

3.1 Perancangan Perangkat Mekanik

Dalam tugas akhir ini akan dirancang suatu prototipe simulasi sistem penyaluran air bersih di desa berbasis RFID (*Radio Frequency Identification*). Menggunakan mikrokontroler *atmega16* ter-integrasi pada *database*. Objek yang akan diamati dalam sistem ini adalah pemanfaatan kartu RW untuk penerima berjenis air bersih. Monitoring dilakukan dengan tujuan dapat mengetahui penyaluran air bersih di desa setiap harinya.

Perancangan perangkat mekanik ini meliputi pembuatan miniatur desa. Di dalam miniatur ini akan diletakkan serangkaian elektronik yang meliputi *board* rangkaian sistem yang meliputi RFID, *LCD*, Mikrokontroler, dan *power supply* untuk seluruh rangkaian. Karena tujuannya untuk memonitoring air bersih yang keluar dari pompa setiap harinya. Miniatur desa ini dibuat dengan papan triplek, besi pembatas, akrilik, hiasan untuk memperindah bentuk miniatur., seperti pada Gambar 3.2:



Gambar 3.2RancanganSistemMekanik

3.2 Perancangan Perangkat Keras

Perancangan perangkat keras terdiri dari rangkaian sistem minimum mikrokontroler AVR *ATMega16* dan RFID *RC522*, *power supply*. Berikut ini dijelaskan mengenai diagram fungsional proses secara keseluruhan, disertai dengan perangkat perencanaan perangkat elektrik secara keseluruhan.

3.2.1 Sistem Minimum Mikrokontroler AVR ATMega16

Dalam prototipe simulasi sistem penyaluran air bersih di desa berbasis RFID (*Radio Frequency Identification*). Menggunakan Mikrokontroler *Atmega16* ter-integrasi pada *database* ini menggunakan dua mikrokontroler sebagai *transmitter* dan *receiver*. Dikarenakan apabila menggunakan satu mikrokontroler *transmitter* dan *receiver* masih kurang dalam pengiriman dan penerimaan data-nya.

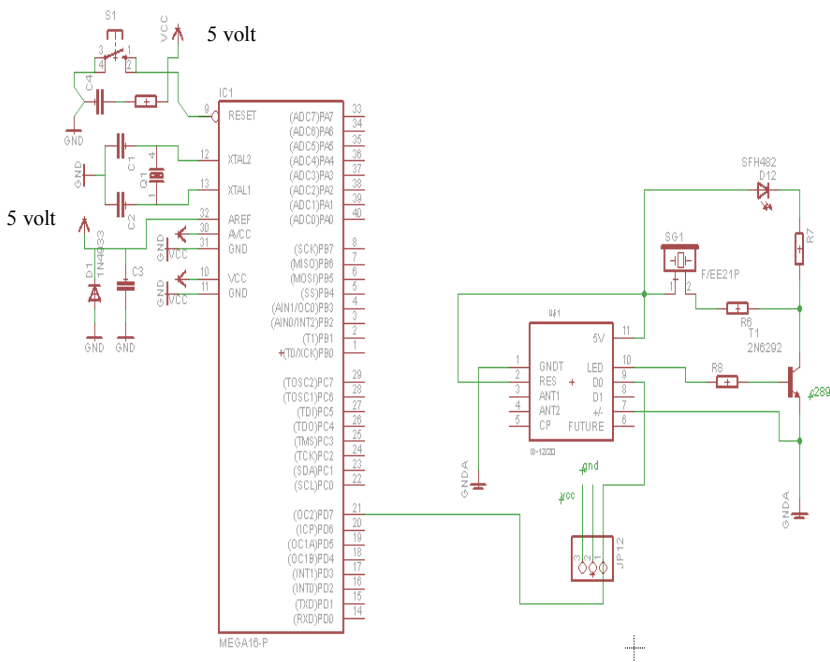
3.2.1.1 Sistem Minimum Mikrokontroler AVR ATMega16 Sebagai Penerima Data RFID

Sistem minimum adalah bagian terpenting dari prototipe simulasi sistem penyaluran air bersih di desa berbasis RFID (*Radio Frequency Identification*). Menggunakan Mikrokontroler *Atmega16* terintegrasi pada *database*. Seperti *microprocessor* pada komputer, minimum sistem adalah otak dari rangkaian untuk mengatur semua perintah eksekusi yang diberikan saat pengisian program.

Mikrokontroler pertama ini memiliki beberapa fungsi utama yaitu untuk menerima data dari *tag* RFID untuk mengetahui *tag* tersebut sesuai dengan *database* yang dimasukkan sebelumnya atau tidak mengirimkan data ke PC. Penggunaan *port I/O* mikrokontroler *ATmega16* adalah sebagai berikut :

PORT D → terhubung dengan RFID *Reader* dan RS-232

1. *PORT D.0* → digunakan untuk menerima data dari RFID *Reader*.
2. *PORT D.1* → digunakan untuk mengirim data dari mikrokontroler ke *Personal Computer*.



Dari Gambar 3.3 dapat dilihat rangkaian sistem minimum dimana terdapat led sebagai indikator mikrokontroler. Selain itu terdapat *push button* sebagai *reset* yang di hubungkan ke *pin 9 (pinreset)* pada mikrokontroler dengan resistor *pull up* dan kapasitor yang berfungsi untuk menghindari *bouncing* atau penekanan *push button* sekali yang di anggap berkali kali oleh mikrokontroler. Pada minimum sistem ini digunakan *X-tal* 12.000 KHz.

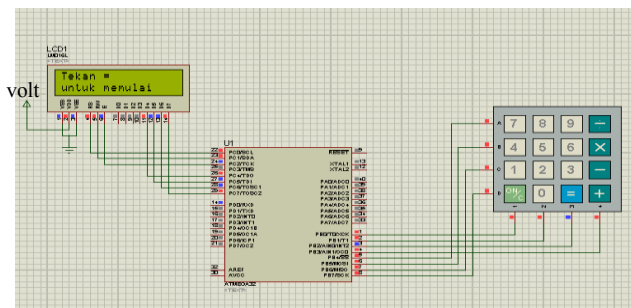
Sistem minimum adalah bagian terpenting dari prototipe simulasi sistem penyaluran air bersih di desa berbasis RFID (*Radio Frequency*

Identification). Menggunakan Mikrokontroler Atmega16 ter-integrasi pada *database*. Seperti *microprocessor* pada komputer, minimum sistem adalah otak dari rangkaian untuk mengatur semua perintah eksekusi yang diberikan saat pengisian program.

Mikrokontroler kedua ini memiliki beberapa fungsi utama untuk menerima data dari *Personal Computer* (PC) untuk memberikan perintah kepada mikrokontroler kedua untuk menjalankan *relay*. Penggunaan *port I/O* mikrokontroler ATmega16 adalah sebagai berikut:

PORT D → terhubung dengan *Personal Computer* (PC).

1. *PORT D.0* → digunakan untuk menerima kiriman perintah dari *Personal Computer* (PC).
2. *PORT D.4* → digunakan untuk mengaktifkan *keypad*.
3. *PORT D.5* → digunakan untuk mengaktifkan LCD.



Gambar 3.4 Simulasi Proteus Kalkulator R-W

Dari Gambar 3.4 terdapat LCD sebagai layar tampilan kalkulator. Selain itu terdapat *keypad* sebagai tombol untuk memasukkan jumlah air bersih yang dikeluarkan.

Pada minimum sistem ini digunakan *X-tal* 12.0000 KHz karena sesuai dengan *datasheet* yang diperlukan oleh mikrokontroler sebagai pendetak.

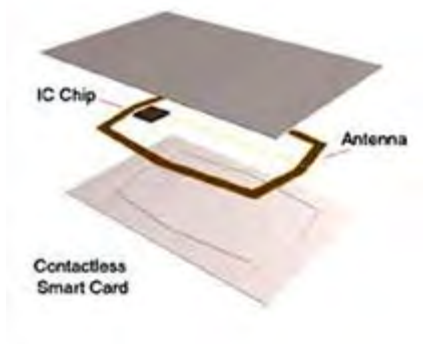
3.2.2 Rangkaian RFID RC 522

Kegunaan RFID adalah untuk memberikan data yang akan diolah oleh mikrokontroler ATmega16. RFID disini adalah bertindak sebagai *receiver*. Kondisi RFID apabila *tag* (kartu)

didekatkan pada *reader*, maka mikokontroler akan menerima data yang di kirim oleh *RFIDreader*. Jika data sudah lengkap maka mikrokontroler akan mengirim *tag* tersebut secara *serial* ke PC.

RFID yang digunakan pada tugas akhir ini adalah RFID RC 522. RFID RC 522 mempunyai kemampuan membaca data dengan frekuensi 13,56mHz. RC 522 dapat bekerja dengan *supply* tegangan 5 VDC 13mA. Komponen ini dihubungkan ke *port* D.0 pada mikrokontroler ATmega16.

Kartu digunakan sebagai identifikasi dari RFID atau RFID *tag*. Dalam *tag* ini terdapat informasi yang digunakan untuk menyimpan informasi berupa identitas dari *tag* itu sendiri. Gambar 3.5 adalah RFID *tag*.



Gambar 3.5 RFID Tag

Format data keluaran yang digunakan adalah tipe ASCII Untuk fungsi tiap – tiap *pin* dan *output* data format dari RFID RC 522 ditunjukkan pada Gambar 3.6 dibawah ini :



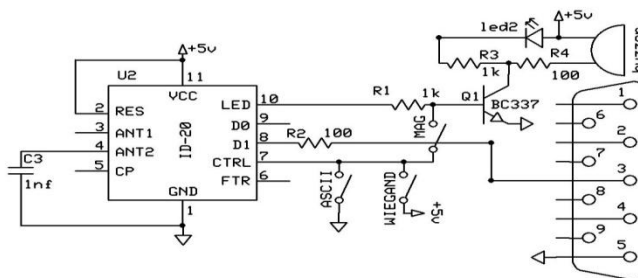
Gambar 3.6RC 522 Pin Out

Deskripsi dan format data *pin* dari RFID RC522 ditunjukkan pada Tabel 3.1 sebagai berikut :

Tabel 3.1 Deskripsi dan Format Data *Output*RC 522

| Pin | Description | ASCII |
|-----|--|---------------------|
| 1 | Zero Volts and Tuning Capacitor Ground | GND 0V |
| 2 | Strap to +5V | Reset |
| 3 | To External Antenna and Tuning Capacitor | Antenna |
| 4 | To External Antenna | Antenna |
| 5 | Card Present | No Function |
| 6 | Future | Future |
| 7 | Format Selector (+/-) | Strap to Ground |
| 8 | Data 1 | CMOS |
| 9 | Data 0 | TTL Data (Inverted) |
| 10 | 3,1 kHz Logic | Beeper/LED |
| 11 | DC Voltage Supply | +5V |

Sedangkan untuk skematik rangkaian RFID yang akan dibuat dapat dilihat pada Gambar 3.7



Gambar 3.7 Skematik Rangkaian RFID

Adapun komponen-komponen yang dibutuhkan untuk membuat rangkaian RFID diatas adalah :

- | | |
|----------------------------|---------|
| 1. RFIDRC 522 | 1 buah. |
| 2. Buzzer 6 Volt | 1 buah. |
| 3. Transistor NPN UTC 8050 | 1 buah. |
| 4. Resistor 1 Ohm | 1 buah. |

- | | |
|--------------------|---------|
| 5. Resistor 1K Ohm | 1 buah. |
| 6. LCD | 1 buah. |
| 7. Pin Header | 1 buah. |

3.3 Perancangan Perangkat Lunak

Berikut ini akan dibahas mengenai perancangan perangkat lunak pada mikrokontroler untuk menerima data yang diperoleh dari RFID dan dikirimkan ke PC lalu PC mengirim perintah ke mikrokontroler dengan menggunakan *software CodeVision AVR C Compiler*.

Mikrokontroler AVR yang digunakan adalah mikrokontroler ATmega16. Program dibuat untuk mendapatkan kode dari RFID. Apabila kode sudah lengkap, yakni dengan mendekatkan RFID *tag* ke *reader*, maka mikrokontroler akan mengirim data tersebut ke PC dan apabila data *tag* RFID sesuai dengan data. Ketika RFID aktif maka input data penyaluran air bersih dapat dijalankan.

3.3.1 Perancangan Perangkat Lunak untuk RFID

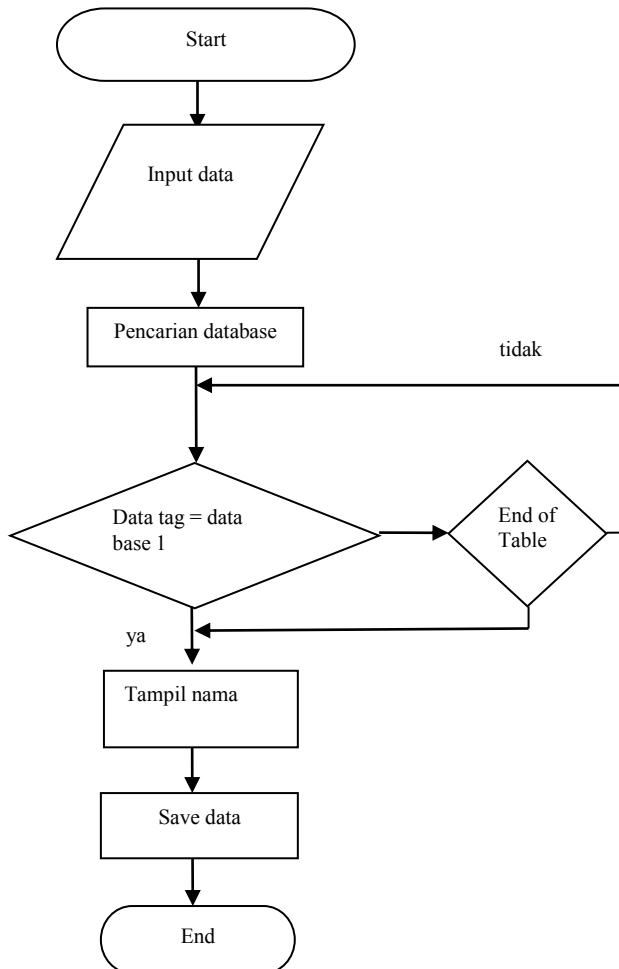
Dalam memrogram RFID ada beberapa hal yang perlu diperhatikan diantaranya :

- | | |
|----------------------------|------------------------|
| 1. <i>Transmitter mode</i> | : <i>Asynchronous.</i> |
| 2. <i>Boud rate</i> | : <i>9.600 bps.</i> |
| 3. <i>Strat bit</i> | : <i>1 bit.</i> |
| 4. <i>Data bit</i> | : <i>8 bit.</i> |
| 5. <i>Stop bit</i> | : <i>1 bit.</i> |
| 6. <i>Parity bit</i> | : <i>non.</i> |
| 7. <i>Control signal</i> | : <i>non.</i> |

RFID disini bertindak sebagai *receiver*. Karena diletakkan di *port D*, maka yang diaktifkan adalah *USART1*. Algoritma pembuatan program untuk RFID ini adalah sebagai berikut :

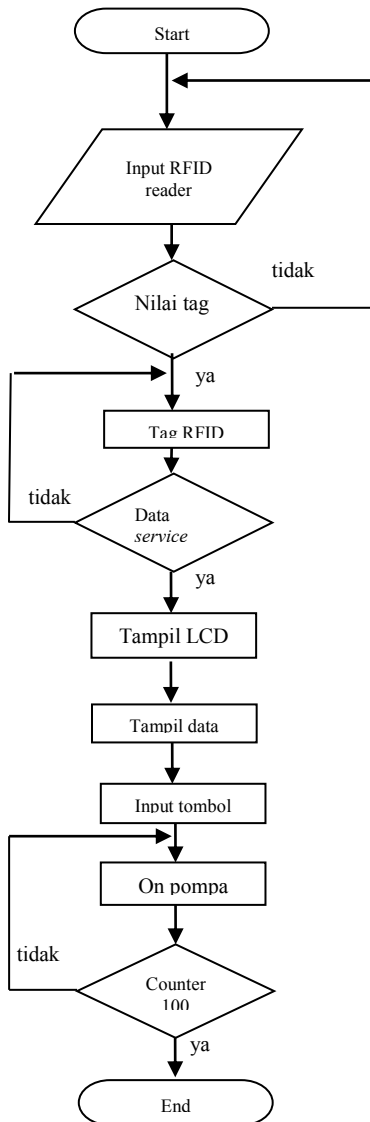
1. RFID dalam kondisi aktif dan siap untuk menerima data.
2. Ketika *tag* di dekatkan ke *reader*, RFID akan membaca data tersebut.
3. Setelah membaca kode – kode dari *tag*, maka akan diteruskan ke mikrokontroler.
4. Setelah ke dibaca mikrokontroler maka kode-kode dari *tag* akan diteruskan ke PC melalui komunikasi serial.

3.3.2 Perancangan Perangkat Lunak untuk Sistem Integrasi PC



Gambar 3.8 *Flowchart* Sistem Perangkat Lunak

3.3.2.1 Perancangan Perangkat Lunak RFID Integrasi PC



Gambar 3.9 *Flow Chart* Sistem RFID dan Database

Dari Gambar 3.8 *flowchart* kinerja sistem secara keseluruhan dapat dijelaskan sebagai berikut :

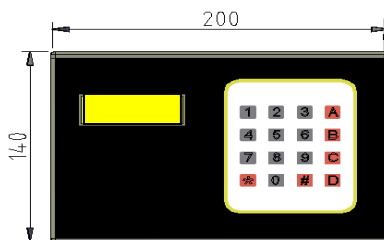
1. Ketika *tag* RFID di dekatkan dengan *reader*, maka kode akan terbaca oleh *reader*.
2. Data dari *reader* tersebut akan di kirim ke mikrokontroler untuk diproses.
3. Mikrokontroler akan mengirim data dari *reader* RFID untuk di kirim ke PC.
4. Apabila data *reader* sama dengan *database* yang ada di PC maka sistem penyaluran air bersih dapat dijalankan.

Dari Gambar 3.9 *flow chart* , kinerja sistem RFID dapat dijelaskan sebagai berikut :

1. Sistem RFID *Reader* mendapat input dari *tag* RFID, RFID *Reader* ini mengirim karakter satu persatu sampai karakternya lengkap ke mikrokontroler.
2. Apabila data RFID sudah sesuai dengan data yang terdapat pada PC, maka mikrokontroler akan memunculkan nama dan alamat di LCD.
3. Selanjutnya operator memasukkan data yang diinginkan pada tombol.
4. Kemudian pompa akan menyala dengan *counter* 100. Maksudnya disini adalah *delay* 100.

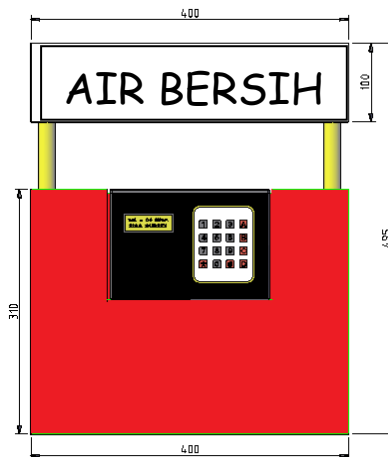
3.4 Perancangan Pembuatan Hardware

Pada pembuatan sistem penyaluran air bersih ini dirancang alat yang digunakan pada RFID *Water* dimana pada perancangannya kami menggunakan *autocad*.



Gambar 3.10 *Hardware RFID Water*

Gambar 3.10 adalah rancangan hardware RFID *Water* sebagai pendeteksi kartu RFID yang telah dimodifikasi menjadi kartu RW yang nantinya akan terintegrasi dengan *database* monitoring air bersih, selain itu hardware RFID *Water* ini mempunyai fungsi kalkulator untuk pembelian air bersih. Dimana kalkulator ini khusus untuk sistem pembelian air bersih.



Gambar 3.11 Perancangan Tempat Peletakan Hardware

Pada Gambar 3.11 adalah tempat dimana hardware RFID *Water* diletakkan pada miniatur desa agar terlihat rapi dan tertata, selain itu desain juga dirancang agar dapat *portable* dimana dapat diubah-ubah dan tidak permanen.

3.5 Perancangan Pembuatan Interface

Rancangan pembuatan *interface* menggunakan *software Visual Studio 2012*. pembuatan *interface* ini merupakan bentuk tampilan yang di rancang untuk sistem pembatasan penerimaan dan monitoring penyaluran air bersih berbasis RFID (*Radio Frequency Identification*) Menggunakan Mikrokontroler Atmega16 terintegrasi pada *database*. Pada sistem *interface* menggunakan *Visual studio 2012* ini dirancang khusus untuk *admin*.

3.5.1 Perancangan Sign Up Page

Rancangan *sign up page* ini merupakan langkah awal *admin* untuk mengakses layar utama pada *interface* menggunakan *Visual Basic* ini. Apabila *admin* tidak mengetahui *password* untuk *sign up page* ini maka tidak bisa mengakses layar utama, seperti pada Gambar 3.12:



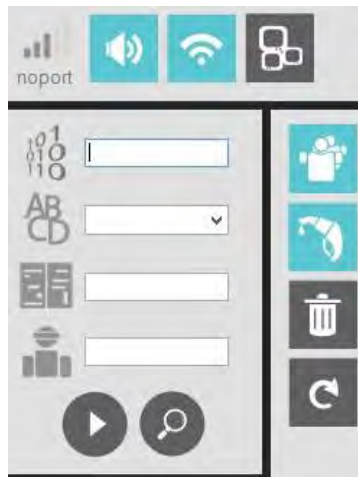
Gambar 3.12 Rancangan Sign Up Page

3.5.2 Perancangan Menu Utama

Rancangan menu utama untuk *admin* ini menyediakan menu-menu yang bisa digunakan sebagai contoh menu registrasi, menu koneksi dll.

3.5.3 Perancangan Menu Koneksi

Rancangan menu koneksi ini merupakan salahsatu bagian dari menu prototipe simulasi sistem penyaluran air bersih di desa berbasis RFID (*Radio Frequency Identification*). Rancangan menu koneksi ini di gunakan untuk menghubungkan *serial* antara rangkaian (*hardware*) dengan PC. Menu koneksi ini dibuat sebagai langkah awal untuk mengaktifkan *interface*prototipe simulasi sistem penyaluran air bersih di desa berbasis RFID (*Radio Frequency Identification*) Menggunakan Mikrokontroler Atmega16 terintegrasi pada *database*, seperti pada Gambar 3.13:



Gambar 3.13Rancangan Menu Koneksi

3.5.4 Perancangan Registrasi

Dalam rancangan registrasi ini dibagi menjadi dua bagian yaitu pengguna dan bukan pengguna.

3.5.4.1 Registrasi Pengguna RFID *Water*

Registrasi pengguna RFID *Water* ini bertujuan untuk memasukkan data pengambilan air bersih. Ketika data penerimaan air

bersih telah dimasukkan maka akan disimpan pada *database* pembelian. Seperti yang ditunjukkan pada Gambar 3.14

Gambar 3.14 *Form Pendaftaran Pengguna RFID Water*

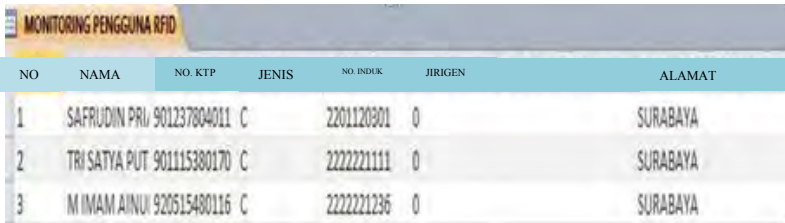
3.5.5 Perancangan *Record* Penduduk

Record penduduk ini bertujuan untuk mengetahui identitas yang tertera pada kartu RW di *database*. Hal ini dilakukan untuk menjaga keamanan data dan keaslian data sehingga tidak perlu membuka *database* *sems access*. seperti pada Gambar 3.15:

Gambar 3.15Rancangan *Record* Pengguna RFID *Water*

3.5.6Perancangan Data Pengguna RFID *Water*

Data penduduk ini bertujuan untuk mengetahui data pengguna dan penerima air bersih. Hal ini agar memudahkan *admin* untuk mengecek pengguna RFID*Water*, seperti yang ditunjukkan pada Gambar 3.16



| NO | NAMA | NO. KTP | JENIS | NO. INDIK | JIRIGEN | ALAMAT |
|----|---------------|--------------|-------|------------|---------|----------|
| 1 | SAFRUDIN PRI | 901237804011 | C | 2201120301 | 0 | SURABAYA |
| 2 | TRI SATYA PUT | 901115380170 | C | 2222221111 | 0 | SURABAYA |
| 3 | M IMAM AINUI | 920515480116 | C | 2222221236 | 0 | SURABAYA |

Gambar 3.16Rancangan data Pengguna RFID *Water*

3.5.7Perancangan Sistem Penggunaan Kartu *Tag* RFIDsebagai ID Pengguna RFID *Water*

Sistem penggunaan kartu *tag RFID* sebagai ID R-W (RFID *Water*) dibagi menjadi beberapa bagian, diantaranya yaitu :

1. Sistem Registrasi RFID *Tag*.
Pengguna RFID *Water* pada saat akan menggunakan kartu *tag* RFID harus memastikan datanya sudah registrasi. Apabila sudah registrasi maka kartu *tag* RFID sudah bisa digunakan.
2. Sistem Penggunaan RFID *Tag*.
RFID *tag*dapat digunakan apabila telah registrasi. Pada saat pengguna RFID *Water*akan memperoleh air bersih, masyarakat cukup mendekatkan RFID *tag* pada RFID *Reader* setelah itu sistem akan bekerja.

3.6 Cara KerjaSistem

RFID*Water*dapatmenjadisolusiuntuk memantau penyaluran air bersih di desa, seperti pada tahap – tahap dibawah ini:

1. Step1

Kondisipadasaatpertama kali penggunaR-Wmelakukanpengisianair bersihpadajirigendenganmenggunakan RFID*Water*akanterlihatpadalayar “Name : mamprut, Jirigen:

0”makapemilikR-Wtersebutdapatmengisidenganair bersihsebanyak 2 Jirigen, apabilapemilikR-Wtersebutinginmengisi 3 Jirigenmakasacaraotomatispenghitunganpompamenjadipenghitunga nharga air bersih yang harus dibeli, seperti pada Gambar 3.17:



Gambar3.17KondisiSaatBelumMengisiJirigen Air Bersih

2. Step 2

KondisipadasaatpenggunaR-Wtelahmengisi 2 Jirigenair bersih akanterlihatpadalayar “ *Name* : mamprut , Jirigen : 2”makapemilikR-Wtersebuttidakdapatmengisidenganair bersih, apabilapemilikR-Wtersebutinginmengisiair bersihsecaraotomatispenghitunganpompamenjadipenghitunganharga air bersih yang harus dibeli, seperti pada Gambar 3.18:



Gambar3.18KondisiSetelahMengisi JirigenAir Bersih

3. Step 3

Selama 24 jam pengguna kondisi R-Wakan berubah menjadi “*Name* : mamprut , *Jirigen* : 0 “. Ketika kondisi ini pengguna RFID-*Water* dapat mengisi kembali jirigennya dengan air bersih.

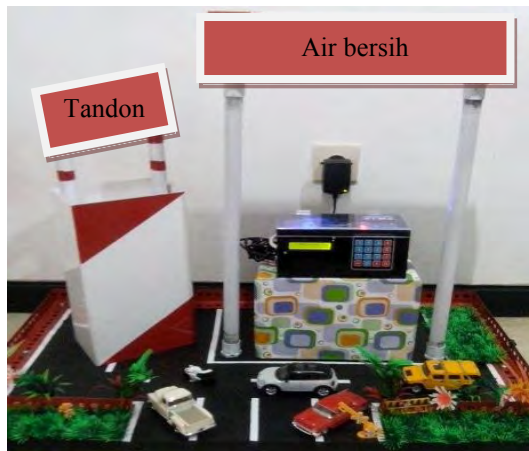
BAB IV

PENGUJIAN DAN ANALISA DATA

Pengujian yang dilakukan pada perangkat keras (*hardware*) ini ditujukan untuk mengetahui tingkat keberhasilan suatu proyek akhir dalam mencapai hasil yang diinginkan. Dengan mengetahui hasil dari pengujian tersebut maka dapat diketahui tingkat keberhasilan suatu alat.

4.1. Desain Miniautur Prototipe Pedesaan

Pada pengaplikasian RFID *Water* ini perlu *prototype* Desa agar dalam simulasinya benar-benar ada dalam suasana desa sehingga pengaplikasian RFID *Water* terlihat seperti kondisi *real*-nya. Pada tampilan awal RFID *Water* akan terlihat tulisan RFID *Water* pada alat penyaluran air bersih sebagai kalimat “*Welcome*” atau ucapan “Selamat Datang” di sistem RFID *Water*. Berikut Gambar 4.1 dan Gambar 4.2 miniatur RFID *Water*:



Gambar 4. 1 Miniatur RFID *Water*

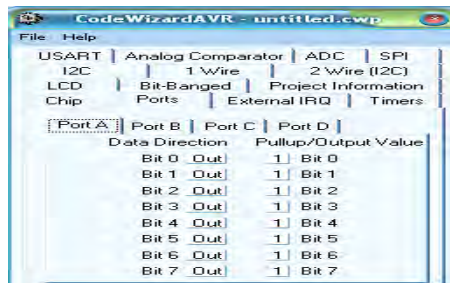


Gambar 4.2 RFID Water Tampak Atas

4.2. Pengujian Rangkaian Mikrokontroler ATmega16

Kami melakukan percobaan untuk mengecek hasil keluaran dari tiap *port* Mikrokontroler dengan cara sebagai berikut:

1. Membuat program menggunakan *software CodeVision AVR C Compiler*.
2. Membuat *project* baru. Gunakan *CodeWizardAVR* untuk menentukan *port – port* yang akan digunakan. Untuk pengisian *logic 1 (high)* maka ganti *Data Direction* menjadi *Out* dan *Pull-up / Output Value* menjadi 1 untuk *Port A*, *Port B*, *Port C*, dan *Port D*. seperti pada Gambar 4.3:



Gambar 4.3 Mengaktifkan *PORT Out* Pada ATMEGA16

3. Setelah itu pilih opsi *File → Generate, Save and Exit* dan beri nama *file* yang sama.
4. Untuk mengecek apakah terdapat peringatan atau *error* pada program tersebut tekan F9.

5. Langkah berikutnya adalah *download* program tersebut ke mikrokontroler. Terlebih dahulu, pastikan bahwa koneksi USB dari PC ke *chip* telah terhubung. Setelah itu tekan *Shift* + F9 lalu pilih *Program to Chip*.
6. Setelah program di *download*, langkah selanjutnya adalah melakukan pengukuran masing – masing pin untuk tiap *port*. Led yang disertakan di *board* mikro ini digunakan sebagai indikator.
7. Nyalakan *power supply*. Indikator led menyala karena mikrokontroler diisi dengan *logic 1 (high)*.
8. Selanjutnya adalah melakukan pengukuran di VCC dan di setiap pin – pin di *PORTA*, *PORTB*, *PORTC*, dan *PORTD* menggunakan avometer digital. Berikut pengukurannya:
 1. Pengukuran tegangan keluaran di *PORT A*
Output Port A dihubungkan dengan kaki positif pada multimeter, dan dilihat berapa *Volt* tegangannya dan kaki *negative* pada multimeter dihubungkan pada *ground*.
 2. Pengukuran tegangan keluaran di *PORT B*
Output Port B dihubungkan dengan kaki positif pada multimeter, dan dilihat berapa *Volt* tegangannya dan kaki *negative* pada multimeter dihubungkan pada *ground*.
 3. Pengukuran tegangan keluaran di *PORT C*
Output Port C dihubungkan dengan kaki positif pada multimeter, dan dilihat berapa *Volt* tegangannya dan kaki *negative* pada multimeter dihubungkan pada *ground*.
 4. Pengukuran tegangan keluaran di *PORT D*
Output Port D dihubungkan dengan kaki positif pada multimeter, dan dilihat berapa *Volt* tegangannya dan kaki *negative* pada multimeter dihubungkan pada *ground*.

Tabel 4.1 Hasil Pengukuran Mikrokontroler Pada *PORT A*

| <i>PORT A</i> | Tegangan <i>Output Low (Volt)</i> | Tegangan <i>Output High (Volt)</i> |
|---------------|-----------------------------------|------------------------------------|
| A.0 | 0,02 | 5,10 |
| A.1 | 0,02 | 5,10 |
| A.2 | 0,01 | 5,10 |
| A.3 | 0,02 | 5,10 |
| A.4 | 0,01 | 5,10 |
| A.5 | 0,01 | 5,10 |
| A.6 | 0,01 | 5,10 |
| A.7 | 0,01 | 5,10 |

Pada Tabel 4.1 dapat dilihat tegangan pada keluaran setiap *Port A* banyak yang berkurang dari tegangan *Vcc* nya, ini dikarenakan tegangan *Vcc* pada saat masuk ke mikrokontroler membagi ke 5 *Port* (atau terkena beban) dimana beban tersebut mempengaruhi nilai masukan pada mikrokontroler ATMega16 yang dapat membuat *error*.

Tabel 4.2 Hasil Pengukuran Mikrokontroler Pada *PORT B*

| <i>PORT B</i> | Tegangan <i>Output Low</i> (Volt) | Tegangan <i>Output High</i> (Volt) |
|---------------|-----------------------------------|------------------------------------|
| B.0 | 0,02 | 4,9 |
| B.1 | 0,02 | 4,9 |
| B.2 | 0,02 | 4,9 |
| B.3 | 0,02 | 4,9 |
| B.4 | 0,02 | 4,9 |
| B.5 | 0,02 | 4,9 |
| B.6 | 0,02 | 4,9 |
| B.7 | 0,02 | 4,9 |

Pada Tabel 4.2 tegangan pada keluaran setiap *Port B* banyak yang berkurang dari tegangan *Vcc* nya, ini dikarenakan tegangan *Vcc* pada saat masuk ke mikrokontroler membagi ke-5 *Port* (atau terkena beban) dimana beban tersebut mempengaruhi nilai masukan pada mikrokontroler ATMega16 yang dapat membuat *error*, hal yang tidak dapat diperkirakan akan tetapi dapat diantisipasi oleh *programmer*.

Tabel 4.3 Hasil Pengukuran Mikrokontroler Pada *PORT C*

| <i>PORT C</i> | Tegangan <i>Output Low</i> (Volt) | Tegangan <i>Output High</i> (Volt) |
|---------------|-----------------------------------|------------------------------------|
| C.0 | 0,02 | 4,9 |
| C.1 | 0,02 | 4,9 |
| C.2 | 0,02 | 4,9 |
| C.3 | 0,02 | 4,9 |
| C.4 | 0,02 | 4,9 |
| C.5 | 0,02 | 4,9 |
| C.6 | 0,01 | 4,9 |
| C.7 | 0,01 | 4,9 |

Pada Tabel 4.3 tegangan pada keluaran setiap *Port C* banyak yang berkurang dari tegangan *Vcc* nya, ini dikarenakan tegangan *Vcc* pada saat masuk ke mikrokontroler membagi ke 5 *Port* (atau terkena beban) dimana beban tersebut mempengaruhi nilai masukan pada mikrokontroler ATMega16 yang dapat membuat *error* hal yang tidak dapat diperkirakan akan tetapi dapat diantisipasi oleh *programmer*.

Tabel 4.4 Hasil Pengukuran Mikrokontroler Pada *PORT D*

| <i>PORT D</i> | Tegangan <i>Output Low</i> (Volt) | Tegangan <i>Output High</i> (Volt) |
|---------------|-----------------------------------|------------------------------------|
| D.0 | 0,02 | 4,9 |
| D.1 | 0,02 | 4,9 |
| D.2 | 0,02 | 4,9 |
| D.3 | 0,02 | 4,9 |
| D.4 | 0,02 | 4,9 |
| D.5 | 0,02 | 4,9 |
| D.6 | 0,02 | 4,9 |
| D.7 | 0,02 | 4,9 |

Pada Tabel 4.4 tegangan pada keluaran setiap *Port D* banyak yang berkurang dari tegangan V_{cc} nya, ini dikarenakan tegangan V_{cc} pada saat masuk ke mikrokontroler membagi ke 5 *Port* (atau terkena beban).

4.3. Pengujian pada RFID RC 522

Kami melakukan pengujian pada RFID RC522 yang dibuat dengan cara menghubungkan RFID ke sebuah komputer dan pengecekkannya melalui *hyperterminal*. Saat jendela *connection description name* akan diisi dengan RFID yang menunjukkan bahwa alat yang dihubungkan adalah RFID seperti pada Gambar 4.4:



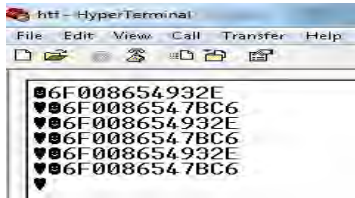
Gambar 4.4 Aplikasi *Hyper Terminal*

setelah tombol *OK* ditekan muncul jendela *connect to*, dan pilih COM2 pada *connecting using*, kemudian tekan tombol *OK*, dan tekan *OK* kembali saat muncul jendela *COM2 properties*. Kemudian setelah muncul jendela akan ditampilkan tampilan *hyper terminal tag* RFID seperti pada Gambar 4.5:



Gambar 4.5 *Connect to COM*

Saat *tag* RFID didekatkan pada *reader* RFID RC522 pada jarak dan kemiringan tertentu maka pada tampilan komputer akan muncul ID seperti pada Gambar 4.6:

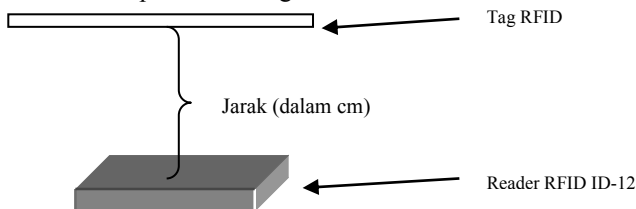


Gambar 4.6 Tampilan Pada *Hyper Terminal* Saat Pengecekan 2 *Tag* RFID

Pada Gambar 4.6 pengujian RFID ini, kami membandingkan jarak dan kemiringan antara *tag* dan *reader* dari RFID, sehingga dapat diketahui jarak dan kemiringan berapakah RFID ini dapat bekerja dan mencari jarak dan posisi yang paling tepat dalam pembacaan RFID.

Pengujian ini menggunakan *tag* dengan nomor seri ” 6F0086547BC6”. Seperti pada Gambar 4.7

- pada kemiringan 0°



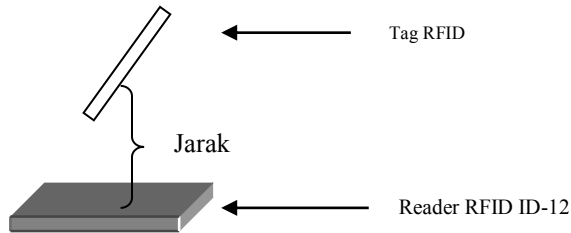
Gambar 4.7 Pengecekan RFID Jarak Kemiringan 0°

Tabel 4.5 Hasil Pengujian Jarak Pada Kemiringan 0°

| Jarak tag dan reader RFID (cm) | RFID mendeteksi/tidak | Pada PC |
|--------------------------------|-----------------------|--------------|
| 8 | Tidak mendeteksi | - |
| 7 | Tidak mendeteksi | - |
| 6 | Tidak mendeteksi | - |
| 5 | Mendeteksi | 6F0086547BC6 |
| 4 | Mendeteksi | 6F0086547BC6 |
| 3 | Mendeteksi | 6F0086547BC6 |
| 2 | Mendeteksi | 6F0086547BC6 |

Dari hasil yang telah diperoleh pada Tabel 4.5 dapat disimpulkan bahwa RFID ID-12 yang digunakan dengan posisi 0° dari *reader* dalam rangkaian kami dapat mendeteksi *tag* dalam *range* jarak 1-5 sentimeter, diluar jarak tersebut (5 cm) *reader* sudah tidak dapat mendeteksi *tag* selanjutnya dapat dilihat pada Gambar 4.8:

- Pada kemiringan 30-45°



Gambar 4.8 Pengecekan RFID Jarak Kemiringan 30° – 45°

Tabel 4.6 Hasil Pengujian Jarak Pada Kemiringan 30° Sampai 45°

| Jarak tag dan reader RFID (cm) | RFID mendeteksi/tidak | Pada PC |
|--------------------------------|-----------------------|--------------|
| 8 | Tidak mendeteksi | - |
| 7 | Tidak mendeteksi | - |
| 6 | Tidak mendeteksi | - |
| 5 | Mendeteksi | 6F0086547BC6 |
| 4 | Mendeteksi | 6F0086547BC6 |
| 3 | Mendeteksi | 6F0086547BC6 |
| 2 | Mendeteksi | 6F0086547BC6 |
| 1 | Mendeteksi | 6F0086547BC6 |

Dari hasil yang telah diperoleh pada Tabel 4.6 dapat disimpulkan bahwa RFID RC 522 yang digunakan dengan posisi 30-45° dari *reader* dalam rangkaian kami dapat mendeteksi *tag* dalam *range* jarak 1-5

sentimeter, diluar jarak tersebut (5 cm) *reader* sudah tidak dapat mendeteksi *tag*. Namun kepekaan RFID menjadi kurang peka jika dibandingkan dengan menggunakan 0°.

4.4 Pengujian Tampilan *Visual Studio 2012* dan *Database*

Untuk memulai program, jalankan program *Visual Studio 2012* dengan klik tombol *start* pada tampilan *visual basic 2012* dan menjalankan programnya melalui program *VB.NET*. seperti pada Gambar 4.9:

| | RFID | SIM | NO_SIM | NAMA | SUBSIDI | PERIODE | WAKTU |
|---|------------|-----|----------|--------|---------|---------|-----------------|
| | e1.45f3.e3 | A | 33220130 | Imam | 15 | 7 | 12/01/2015 9:15 |
| ▶ | e2.45f1.e4 | C | 22049381 | zainal | 5 | 7 | 19/01/2015 9:15 |
| * | | | | | | | |

Gambar 4.9 *Database VB.NET*

4.4.1 Tampilan Layar Implementasi Program *Database*

1. Loading Pembukaan *Database*

Sebelum mengakses *database* muncul tampilan *loading* program untuk menampilkan *log in form* yang akan diakses untuk operasi sistem *database*.

Dengan adanya *loading* program operator dapat mempersiapkan data-data yang akan di masukkan kedalam *database* dan merupakan identitas suatu program *database*.

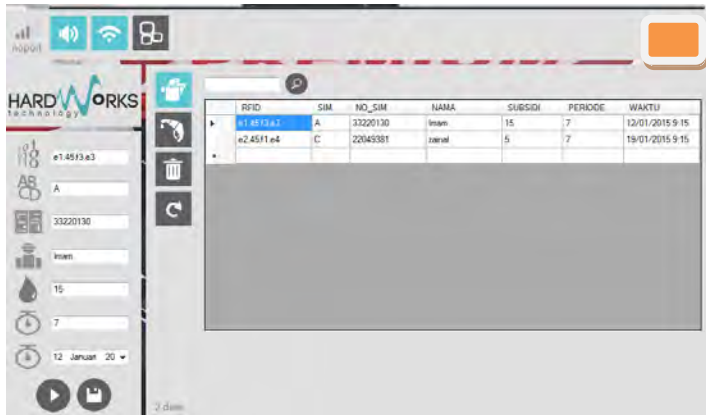
2. Layar *Login Form*

Sebelum mengakses tampilan ini, pengguna diharuskan untuk mempunyai *account* yang digunakan untuk *login*. Jika pengguna tidak mempunyai akun yang benar maka orang tersebut tidak bisa menjalankan program ini. Seperti pada Gambar 4.10:



Gambar 4.10 Tampilan Layar *Log in Form*

3. Setelah melakukan *login* maka akan tampil tampilan utama seperti yang ditunjukkan pada Gambar 4.11 berikut:



Gambar 4.11 Tampilan Utama

Pada tampilan utama ini terdapat 3 item utama yaitu koneksi, regrestasi, *record*. Pada menu terdapat beberapa bagian yaitu registrasi, *record* dan *search* seperti pada Gambar 4.12:



Gambar 4.12 Tampilan Konektivitas



Gambar 4.13 *Log in Form* Pengisian COM Untuk Konektivitas

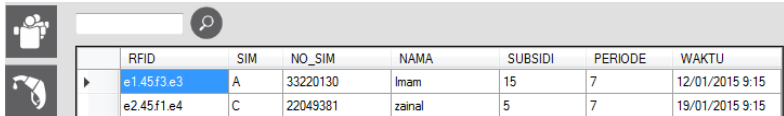
Pada tampilan koneksi dari Gambar 4.13 *user* harus memasukkan data *port* yang telah tersambung pada PC. Setelah itu tekan tombol *connect* pada tampilan koneksi. Maka selanjutnya *serial* telah tersambung dengan PC. Misal konek untuk COM 10 maka *port* yang harus dipilih untuk konektivitas dengan PC adalah COM 10 begitu juga selanjutnya ketika COM 9 maka dipilih COM 9 untuk konektivitas dengan PC.

Gambar 4.14 Tampilan Registrasi Pengguna RFID *Water*

Pada Gambar 4.14 adalah *form* untuk mendaftarkan pengguna RFID *Water* Pendaftaran dilakukan dengan urutan : SN RFID, jenis kartu, No. kartu, Nama pengguna RFID *Water*, Banyak air yang di

berikan untuk Pengguna RFID *Water*, Jangkah waktu (hari) jika tertulis 7 maka jangka waktu 7 hari, Waktu untuk menentukan awal pembuatan R-W.

Setelah melakukan pendaftaran maka klik tombol *save* untuk menyimpan dan tombol *play* untuk melanjutkan pendataan pengguna R-W selanjutnya, seperti pada Gambar 4.15:



| | RFID | SIM | NO_SIM | NAMA | SUBSIDI | PERIODE | WAKTU |
|---|------------|-----|----------|--------|---------|---------|-----------------|
| ▶ | e1.45f3 e3 | A | 33220130 | Imam | 15 | 7 | 12/01/2015 9:15 |
| | e2.45f1 e4 | C | 22049381 | zainal | 5 | 7 | 19/01/2015 9:15 |

Gambar 4.15 Tampilan Data Setelah Terdaftar

4.5. Pengujian RFID *Water*

Untuk pengujian R-W yang digunakan sebagai LCD sebagai tampilan awal bertuliskan “RFID *Water*” setelah dimasukkan, sebagai indikator aktifnya mikro dan RC 522, seperti pada Gambar 4.16:



Gambar 4.16 Tampilan Bertuliskan RFID *Water*

Pada saat program dijalankan atau pada saat *tag* RFID aktif maka otomatis sistem berjalan dengan tampilan awal setelah *tag* RFID adalah masukan Jirigen/rupiah. Bisa dilihat pada Gambar 4.17:



Gambar 4.17 Pilihan Jirigen/Rupiah.

4.6. Pengujian Sistem Keseluruhan

Pengujian sistem secara keseluruhan dilakukan untuk menguji bagaimana sistem yang dirancang dapat berjalan sesuai dengan yang diharapkan. langkah langkah simulasi ini dapat dilakukan dengan :

1. Adanya Perbedaan saat awal perancangan dan simulasi keseluruhan, pada saat awal perancangan terjadi ketidak stabilan inputan dengan sama-sama kondisi 3 jirigen. Seperti pada Gambar 4.18:



(a) Kondisi Awal Perancangan



(b) Kondisi Simulasi Sistem

Gambar 4.18 Perbedaan 2 Kondisi Awal dan Kondisi Simulasi Sistem Keseluruhan

Dengan adanya perbedaan kondisi maka kami stabilkan dengan kondisi kedua *input* sama disaat pengambilan 3 jirigen.

2. Ketika *tag* RFID di dekatakan pada *reader* RFID seperti Gambar 4.18 maka *reader* RFID akan membaca *tag* RFID tersebut. Akan muncul tampilan nama dan no pengguna, dengan sistem integrasi *database* maka awal *tag* RFID menjadi awal inputan *database* apabila kartu tersebut terdaftar sebagai pengguna R-W dengan mendapatkan jatah kuota yang telah ditentukan. Gambar 4.19 menunjukkan hasil tampilan awal ketika setelah melakukan *tag* RFID.



Gambar 4.19 *Tag* RFID dan *Reader* RFID

3. Ketika *tag* RFID sudah di dekatkan pada *reader* RFID maka *input* nilai jirigen untuk menerima air bersih yang di inginkan dengan menekan tombol yang tersedia pada keypad, untuk mendapatkan 3 jirigen maka harus menekan tombol “003” kemudian akan muncul pada LCD dengan tulisan angka jirigen yang di inginkan yaitu “003” seperti Gambar 4.19. Pembelian dibawah jumlah jirigen yang di tentukan akan mendapatkan harga subsidi dan ketika lebih dari jirigen yang ditentukan maka harga secara otomatis menginjak angka pembelian, oleh karena itu nantinya akan ada informasi sisa air yaitu : jirigen

yang ditentukan – penerima air bersih yang dapat dilihat pada Gambar 4.20:



Gambar 4.20 Nilai Rupiah Pembelian

4. Setelah pembelian maka akan masuk data simpanan pembelian pada *database*.
5. Setelah data tersimpan dan keluar rupiah yang diinginkan dari pembelian 3 jirigen maka selanjutnya ditekan tombol C untuk mengeluarkan air pada jirigen dan rupiah hingga 0 serta diikuti dengan keluaran air yang didapat Seperti Gambar 4.21:



Gambar 4.21 Counter dan Keluaran Air

6. Pada saat *counter* mati keluaran air akan juga akan mati, kemudian pengguna akan mengetahui sisa penggunaan air yang

akan ditampilkan pada LCD dan *database* sistem monitoring penyaluran air bersih. Seperti Gambar 4.22:



Gambar 4.22 Status Hasil Sisa Penyaluran Air

{Halaman ini sengaja dikosongkan}

BAB V

PENUTUP

Dari perancangan dan pengujian alat, dapat ditarik kesimpulan dan saran mengenai tugas akhir berikut :

5.1. Kesimpulan

- a) *Reader* RFID dapat membaca *tag* secara maksimal pada jarak 3 cm, dan akan mendeteksi secara maksimal ketika pengecekan *tag* dilakukan dengan sudut 0^0 terhadap *reader*
- b) Penyimpanan catatan pada *database* hanya di fokuskan pada *tag* RFID dan juga pengguna apakah memanfaatkan atau tidak.
- c) Sistem hanya diterapkan pada pengguna RFID *Water*.
- d) Sistem kerja akan *reset* pada waktu yang ditentukan.

5.2. Saran

- a) Saran yang dapat diberikan untuk pengembangan sistem ini selanjutnya adalah ketika melakukan pengecekan jangan menggunakan *tag* yang berbeda secara bersamaan
- b) Untuk program *Visual Basic* sebaiknya selalu dilihat tiap hari karena biasanya mudah *error*
- c) Sistem ini terfokus pada pengamanan dengan RFID yang dapat di kembangkan lagi dengan penambahan *server* sehingga RFID *Water* dapat terintegrasi dengan baik dan tepat sasaran

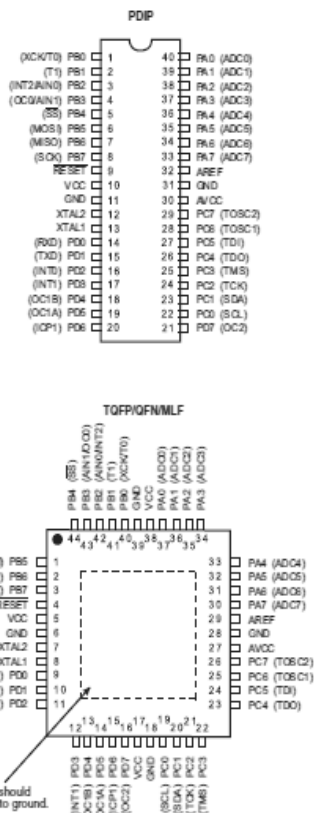
{Halaman ini sengaja dikosongkan}

LAMPIRAN 1

ATmega16(L)

Pin Configurations

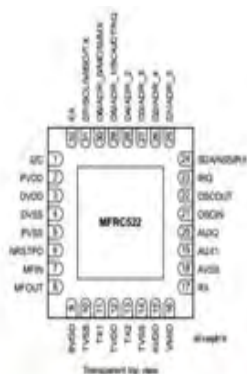
Figure 1. Pinout ATmega16



Disclaimer

Typical values contained in this datasheet are based on simulations and characterization of other AVR microcontrollers manufactured on the same process technology. Min and Max values will be available after the device is characterized.





| Communication direction | Signal type | Transfer speed | | | |
|---|------------------------|----------------------------|----------------------------|----------------------------|----------------------------|
| | | 106 kBd | 212 kBd | 424 kBd | 848 kBd |
| Reader to card (send data from the MFRC522 to a card) | reader side modulation | 100 % ASK | 100 % ASK | 100 % ASK | 100 % ASK |
| | bit encoding | modified Miller encoding | modified Miller encoding | modified Miller encoding | modified Miller encoding |
| | bit length | 128 (13.56 μ s) | 64 (13.56 μ s) | 32 (13.56 μ s) | 16 (13.56 μ s) |
| Card to reader (MFRC522 receives data from a card) | card side modulation | subcarrier load modulation | subcarrier load modulation | subcarrier load modulation | subcarrier load modulation |
| | subcarrier frequency | 13.56 MHz / 16 | 13.56 MHz / 16 | 13.56 MHz / 16 | 13.56 MHz / 16 |
| | bit encoding | Manchester encoding | BPSK | BPSK | BPSK |

| Pin | Interface type | | |
|-----|----------------|--------------|--------------|
| | UART (input) | SPI (output) | PC-bus (I/O) |
| SDA | RX | NSS | SDA |
| I2C | 0 | 0 | 1 |
| EA | 0 | 1 | EA |
| D7 | TX | MISO | SCL |
| D6 | MX | MOSI | ADR_0 |
| D5 | DTRQ | SCK | ADR_1 |
| D4 | - | - | ADR_2 |
| D3 | - | - | ADR_3 |
| D2 | - | - | ADR_4 |
| D1 | - | - | ADR_5 |

❖ Program RFID READER :

```

/*
 * MFRC522.cpp
 *
 * Created on: 2013-3-26
 * Author: shawn
 */

#include "MFRC522.h"
#include <mega32a.h>
#include <spi.h>
// #include <Arduino.h>

#define MAX_LEN 16

//-----MFRC522 register -----
//Page 0:Command and Status
#define Reserved00 0x00
#define CommandReg 0x01
#define CommIEnReg 0x02
#define DivIEnReg 0x03
#define CommIrqReg 0x04
#define DivIrqReg 0x05
#define ErrorReg 0x06
#define Status1Reg 0x07
#define Status2Reg 0x08
#define FIFODataReg 0x09
#define FIFOLevelReg 0x0A
#define WaterLevelReg 0x0B
#define ControlReg 0x0C
#define BitFramingReg 0x0D
#define CollReg 0x0E
#define Reserved01 0x0F

//Page 1:Command
#define Reserved10 0x10
#define ModeReg 0x11
#define TxModeReg 0x12
#define RxModeReg 0x13
#define TxControlReg 0x14

```



```

#define TxAutoReg      0x15
#define TxSelReg       0x16
#define RxSelReg       0x17
#define RxThresholdReg 0x18
#define DemodReg       0x19
#define Reserved11     0x1A
#define Reserved12     0x1B
#define MifareReg      0x1C
#define Reserved13     0x1D
#define Reserved14     0x1E
#define SerialSpeedReg 0x1F

```

//Page 2:CFG

```

#define Reserved20      0x20
#define CRCResultRegM  0x21
#define CRCResultRegL  0x22
#define Reserved21     0x23
#define ModWidthReg    0x24
#define Reserved22     0x25
#define RFCfgReg       0x26
#define GsNReg         0x27
#define CWGsPReg       0x28
#define ModGsPReg      0x29
#define TModeReg       0x2A
#define TPrescalerReg  0x2B
#define TReloadRegH    0x2C
#define TReloadRegL    0x2D
#define TCounterValueRegH 0x2E
#define TCounterValueRegL 0x2F

```

//Page 3:TestRegister

```

#define Reserved30      0x30
#define TestSel1Reg     0x31
#define TestSel2Reg     0x32
#define TestPinEnReg    0x33
#define TestPinValueReg 0x34
#define TestBusReg      0x35
#define AutoTestReg     0x36
#define VersionReg      0x37
#define AnalogTestReg   0x38

```

```

#define TestDAC1Reg      0x39
#define TestDAC2Reg      0x3A
#define TestADCReg       0x3B
#define Reserved31       0x3C
#define Reserved32       0x3D
#define Reserved33       0x3E
#define Reserved34       0x3F

//-----
/*
//4 bytes Serial number of card, the 5 bytes is verify bytes
uchar writeData[16] = {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100 };
//initialize to 100 USD

//buffer A password, 16 buffer, the password of every buffer is 6 byte
uchar sectorKeyA[16][16] = {
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF },
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF },
//{0x19, 0x84, 0x07, 0x15, 0x76, 0x14},
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF },
};

uchar sectorNewKeyA[16][16] = {
{0xFF, 0xFF, 0xFF, 0xFF, 0xFF, 0xFF },
{0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xff, 0x07,
0x80, 0x69, 0x19, 0x84,
0x07, 0x15, 0x76, 0x14 },
//you can set another key , such as " 0x19, 0x84, 0x07, 0x15, 0x76,
0x14 "
//{0x19, 0x84, 0x07, 0x15, 0x76, 0x14, 0xff,0x07,0x80,0x69,
0x19,0x84,0x07,0x15,0x76,0x14},
// but when loop, please set the sectorKeyA, the same key, so that RFID
module can read the card
{0xFF, 0xFF, 0xFF, 0xFF,
0xFF, 0xFF, 0xff, 0x07,
0x80, 0x69, 0x19, 0x33,
0x07, 0x15, 0x34, 0x14},
};

```

❖ Program RFID RC 522 Conect To Microcontroler :

```
/*
 * MFRC522.h - Ported from https://github.com/ghalfacree/Arduino-
Sketches/blob/master/RFID2560/RFID2560.pde
 *
 * Created on: 2013-3-26
 * Author: shawn
 */

#ifndef MFRC522_H_
#define MFRC522_H_

#define uchar unsigned char
#define uint unsigned int

//MF522 command bits
#define PCD_IDLE 0x00 //NO action; cancel current
commands
#define PCD_AUTHENT 0x0E //verify password key
#define PCD_RECEIVE 0x08 //receive data
#define PCD_TRANSMIT 0x04 //send data
#define PCD_TRANSCEIVE 0x0C //send and receive data
#define PCD_RESETPHASE 0x0F //reset
#define PCD_CALCCRC 0x03 //CRC check and caculation

//Mifare_One card command bits
#define PICC_REQIDL 0x26 //Search the cards that not
into sleep mode in the antenna area
#define PICC_REQALL 0x52 //Search all the cards in the
antenna area
#define PICC_ANTICOLL 0x93 //prevent conflict
#define PICC_SEIECTTAG 0x93 //select card
#define PICC_AUTHENT1A 0x60 //verify A password key
#define PICC_AUTHENT1B 0x61 //verify B password key
#define PICC_READ 0x30 //read
#define PICC_WRITE 0xA0 //write
#define PICC_DECREMENT 0xC0 //deduct value
#define PICC_INCREMENT 0xC1 //charge up value
#define PICC_RESTORE 0xC2 //Restore data into buffer
```

```

#define PICC_TRANSFER          0xB0    //Save data into buffer
#define PICC_HALT              0x50    //sleep mode

//The mistake code that return when communicate with MF522
#define MI_OK                  0
#define MI_NOTAGERR           1
#define MI_ERR                 2

//class MFRC522 {
//public:
//    MFRC522(int PinSS, int PinRST);
//    virtual ~MFRC522();
//    void MFRC522_Reset(void);
//    void MFRC522_Init(void);
//    unsigned char MFRC522_Request(unsigned char reqMode,
unsigned char *TagType);
//    unsigned char MFRC522_Anticoll(unsigned char *serNum);
//    unsigned char MFRC522_SelectTag(unsigned char *serNum);
//    unsigned char MFRC522_Auth(unsigned char authMode,
unsigned char BlockAddr, unsigned char *Sectorkey, unsigned char
*serNum);
//    unsigned char MFRC522_Read(unsigned char blockAddr,
unsigned char *recvData);
//    unsigned char MFRC522_Write(unsigned char blockAddr,
unsigned char *writeData);
//    unsigned char MFRC522_Halt(void);
//private:
//    void Write_MFRC522(unsigned char addr, unsigned char val);
//    unsigned char Read_MFRC522(unsigned char addr);
//    unsigned char MFRC522_ToCard(unsigned char command,
unsigned char *sendData, unsigned char sendLen, unsigned char
*backData, unsigned int *backLen);
//    void MFRC522_CalculateCRC(unsigned char *pIndata,
unsigned char len, unsigned char *pOutData);
//    void MFRC522_SetBitMask(unsigned char reg, unsigned char
mask);
//    void MFRC522_ClearBitMask(unsigned char reg, unsigned
char mask);
//    void MFRC522_AntennaOn(void);
//    void MFRC522_AntennaOff(void);

```

```

unsigned char MFRC522_getFirmwareVersion();

    //int _pinReset;
    //int _pinSlaveSelect;
//};

#endif /* MFRC522_H_ */

```

❖ Program Keypad :

```

//unsigned char digit2_in (unsigned char input);
unsigned int digit3_in (unsigned int input);

// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void) // intrup timer 20ms
{
    TCNT0=0x28; // intrup timer 20ms
    if(flag_sn) { timer_sn++; };
    if(flag_go) {if(++timer_ltr>8) { contr_ltr++; timer_ltr=0; }; };
    if(++cek_data==5) { cek_data=0; // scening keypad tiap( 20x5
    )ms
    PORTA =0xF0; if((PINA & 0xF0) !=0xF0) {
    PORTA = 0xFF; PORTA.0 =0; delay_ms(5);
    switch ( PINA ) {
        case 0x7E : key = 'A'; flag_key =1;
            break;
        case 0xBE : key = '3'; flag_key =1;
            break;
        case 0xDE : key = '2'; flag_key =1;
            break;
        case 0xEE : key = '1'; flag_key =1;
            break;
    };
    PORTA = 0xFF; PORTA.1 =0; delay_ms(5);
    switch ( PINA ) {
        case 0x7D : key = 'B'; flag_key =1;
            break;
        case 0xBD : key = '6'; flag_key =1;
            break;
    };
    };
    };
}

```

```

    case 0xDD : key = '5'; flag_key=1;
                break;
    case 0xED : key = '4'; flag_key=1;
                break;
    };
    PORTA = 0xFF; PORTA.2 =0; delay_ms(5);
    switch ( PINA ) {
        case 0x7B : key = 'C'; flag_key=1;
                    break;
        case 0xBB : key = '9'; flag_key=1;
                    break;
        case 0xDB : key = '8'; flag_key=1;
                    break;
        case 0xEB : key = '7'; flag_key=1;
                    break;
    };
    PORTA = 0xFF; PORTA.3 =0; delay_ms(5);
    switch ( PINA ) {
        case 0x77 : key = 'D'; flag_key=1;
                    break;
        case 0xB7 : key = '#'; flag_key=1;
                    break;
        case 0xD7 : key = '0'; flag_key=1;
                    break;
        case 0xE7 : key = '*'; flag_key=1;
                    break;
    };
    PORTA = 0xFF;
};
};
}

```

Keypad 4x4

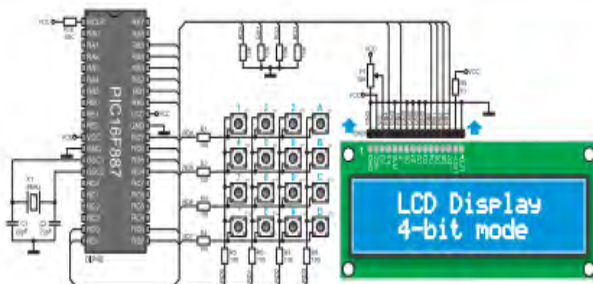


Figure 4: Keypad, LCD and microcontroller connection schematic

Example 1: Program written in mikroC PRO for PIC

```

unsigned short kp, cnt, sdata1 = 0;
char buf[5];

char keypadPort at PORTD;

sbit LCD_RS at RB4_bit;
sbit LCD_EN at RB5_bit;
sbit LCD_D4 at RB0_bit;
sbit LCD_D5 at RB1_bit;
sbit LCD_D6 at RB2_bit;
sbit LCD_D7 at RB3_bit;

sbit LCD_RS_Direction at TRISB4_bit;
sbit LCD_EN_Direction at TRISB5_bit;
sbit LCD_D4_Direction at TRISB0_bit;
sbit LCD_D5_Direction at TRISB1_bit;
sbit LCD_D6_Direction at TRISB2_bit;
sbit LCD_D7_Direction at TRISB3_bit;

void main() {
    cnt = 0;
    Keypad_Init();
    ANSEL = 0;
    ANSELH = 0;

    Lcd_Init();
    Lcd_Cmd(LCD_CLEAR);
    Lcd_Cmd(LCD_CURSOR_OFF);
    Lcd_Out(1, 1, " ");
    Lcd_Out(1, 1, "Key ");
    Lcd_Out(2, 1, "Times");

    do {
        kp = 0;

        // Wait for key to be pressed and released
        do
            kp = Keypad_Key_Click();
        while (kp);

        // Prepare value for output, transform key to it's ASCII value
        switch (kp) {
            //case 10: kp = 42; break; // *
            //case 11: kp = 48; break; // 0
            //case 12: kp = 35; break; // #
            //default: kp = 48;

            case 1: kp = 49; break; // 1
            case 2: kp = 50; break; // 2
            case 3: kp = 51; break; // 3
            case 4: kp = 65; break; // A
            case 5: kp = 52; break; // 4
            case 6: kp = 53; break; // 5
            case 7: kp = 54; break; // 6
            case 8: kp = 66; break; // B
            case 9: kp = 55; break; // 7
            case 10: kp = 56; break; // 8
            case 11: kp = 57; break; // 9
            case 12: kp = 67; break; // C
            case 13: kp = 42; break; // *
            case 14: kp = 48; break; // 0
            case 15: kp = 35; break; // #
            case 16: kp = 68; break; // D
        }

        // If key != sdata1
        if (kp != sdata1) {
            cnt = 1;
            sdata1 = kp;

            // If pressed key is same as previous
            while (
                cnt++
            ) {
                Lcd_Out(1, 10, kp);

                // Print key ASCII value on LCD

                // If counter variable overflow
                if (cnt == 255) {
                    cnt = 0;
                    Lcd_Out(2, 10, " ");

                    // Transform counter value to string
                    // Display counter value on LCD
                    WordToStr(cnt, buf);
                    Lcd_Out(2, 10, buf);
                }
            }
        }
    } while (1);
}

```

LAMPIRAN 2

1. Atmel



1. Pin Configurations

Figure 1-1. Pinout ATmega48A/PA/88A/PA/168A/PA/328P

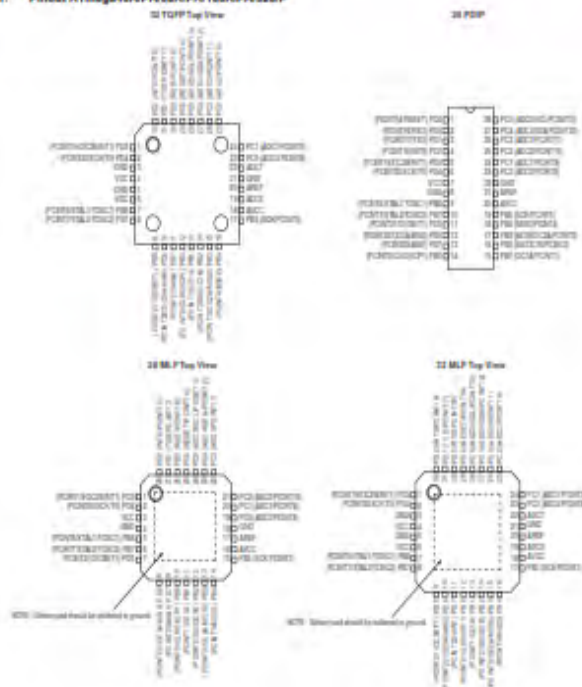


Table 1-1. 32UFPGA - Pinout ATmega48A/88A/88PA/168A/168PA

| | 1 | 2 | 3 | 4 | 5 | 6 |
|---|-----|-----|-----|------|------|------|
| A | PD2 | PD1 | PC6 | PC4 | PC2 | PC1 |
| B | PD3 | PD4 | PD6 | PC5 | PC3 | PC0 |
| C | GND | GND | | ADCT | GND | |
| D | VDD | VDD | | | AREP | AD06 |
| E | PB6 | PD6 | PB0 | PB2 | AVDD | PB5 |
| F | PB7 | PD6 | PD7 | PB1 | PB3 | PB4 |

1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the Inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the Inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..5 is used as TOSC2...1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 62 and "System Clock and Clock Options" on page 27.

1.1.4 Port C (PC3:0)

Port C is a 4-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC3...0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/RESET

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 29-11 on page 305. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 65.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 66.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6...4 use digital supply voltage, V_{CC} .

1.1.8 $AREF$

$AREF$ is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

2. Datasheet IC MAX232

Product Folder
 Sample & Buy
 Technical Documents
 Tools & Software
 Support & Community

MAX232, MAX232E
SEMI-CONDUCTOR, INTEGRATED CIRCUIT, LOGIC, BUFFER, DRIVER, RECEPTOR

MAX232x Dual EIA-232 Drivers/Receivers

1 Features

- Meets or Exceeds TIA/EIA-232-F and ITU Recommendation V.28
- Operates From a Single 5-V Power Supply With 1.0- μ F Charge-Pump Capacitors
- Operates up to 120 kbit/s
- Two Drivers and Two Receivers
- ± 30 -V Input Levels
- Low Supply Current: 8 mA Typical
- ESD Protection Exceeds JESD 22
 - 2000-V Human-Body Model (A114-A)
- Upgrade With Improved ESD (15-kV HBM) and 0.1- μ F Charge-Pump Capacitors is Available With the MAX202 Device

3 Description

The MAX232 device is a dual driver/receiver that includes a capacitive voltage generator to supply TIA/EIA-232-F voltage levels from a single 5-V supply. Each receiver converts TIA/EIA-232-F inputs to 5-V TTL/CMOS levels. These receivers have a typical threshold of 1.3 V, a typical hysteresis of 0.5 V, and can accept ± 30 -V inputs. Each driver converts TTL/CMOS input levels into TIA/EIA-232-F levels.

Device Information⁽¹⁾

| ORDER NUMBER | PACKAGE (PIN) | BODY SIZE |
|--------------|---------------|---------------------------|
| MAX232x | SOIC (16) | 9.40 mm \times 3.91 mm |
| | SOIC (16) | 10.30 mm \times 7.50 mm |
| | SSOP (16) | 10.20 mm \times 6.35 mm |
| | SSOP (16) | 10.2 mm \times 5.30 mm |

(1) For all available packages, see the orderable addresses at the end of the datasheet.

2 Applications

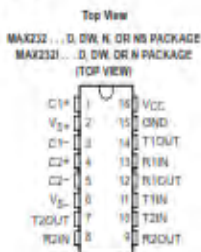
- TIA/EIA-232-F
- Battery-Powered Systems
- Terminals
- Modems
- Computers

4 Simplified Schematic



An IMPORTANT NOTICE at the end of this data sheet addresses availability, warranty, changes, use in safety-critical applications, intellectual property matters and other important disclosures. PRODUCTION DATA.

6 Pin Configuration and Functions



Pin Functions

| PIN | | TYPE | DESCRIPTION |
|--------------|--------|------|--|
| NAME | NO. | | |
| C1+ | 1 | — | Positive lead of C1 capacitor |
| VS+ | 2 | O | Positive charge pump output for storage capacitor only |
| C1- | 3 | — | Negative lead of C1 capacitor |
| C2+ | 4 | — | Positive lead of C2 capacitor |
| C2- | 5 | — | Negative lead of C2 capacitor |
| VS- | 6 | O | Negative charge pump output for storage capacitor only |
| T2OUT, T2OUT | 7, 14 | O | RS232 line data output (to remote RS232 system) |
| R2IN, R1IN | 8, 13 | I | RS232 line data input (from remote RS232 system) |
| R2OUT, R1OUT | 9, 12 | O | Logic data output (to UART) |
| T2IN, T1IN | 10, 11 | I | Logic data input (from UART) |
| GND | 15 | — | Ground |
| VCC | 16 | — | Supply Voltage. Connect to external 5V power supply |

MAX232, MAX232I

D140191 – FEBRUARY 1995 – REVISED NOVEMBER 2014

www.ti.com
7 Specifications
7.1 Absolute Maximum Ratings⁽¹⁾

over operating free-air temperature range (unless otherwise noted)

| | | MIN | MAX | UNIT |
|----------|---|----------------------------------|--|------|
| V_{CC} | Input supply voltage range ⁽²⁾ | -0.3 | 0 | V |
| V_{OH} | Positive output supply voltage range | $V_{CC} - 0.3$ | 15 | V |
| V_{OL} | Negative output supply voltage range | -0.3 | -15 | V |
| V_I | Input voltage range | $T1IN, T2IN$ $R1IN, R2IN$ | -0.3 $V_{CC} + 0.3$ | V |
| V_O | Output voltage range | $T1OUT, T2OUT$ $R1OUT, R2OUT$ | $V_{OH} - 0.3$ $V_{OL} + 0.3$ -0.3 $V_{CC} + 0.3$ | V |
| | Short-circuit duration | $T1OUT, T2OUT$ | Unlimited | |
| T_J | Operating virtual junction temperature | | 150 | °C |

(1) Stresses beyond those listed under Absolute Maximum Ratings may cause permanent damage to the device. These are stress ratings only, and functional operation of the device at these or any other conditions beyond those indicated under Recommended Operating Conditions is not implied. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

(2) All voltages are with respect to network GND.

7.2 Handling Ratings

| | | MIN | MAX | UNIT | |
|-----------|---------------------------|--|-----|------|---|
| T_{stg} | Storage temperature range | -65 | 150 | °C | |
| V_{ESD} | Electrostatic discharge | Human body model (HBM), per ANSI/ESDA/JEDEC J5-001, all pins ⁽¹⁾ | 0 | 2000 | V |
| | | Charged device model (CDM), per JEDEC specification JESD22-C101, all pins ⁽²⁾ | 0 | 1000 | |

(1) JEDEC document JEP155 states that 500-V HBM allows safe manufacturing with a standard ESD control process.

(2) JEDEC document JEP157 states that 250-V CDM allows safe manufacturing with a standard ESD control process.

7.3 Recommended Operating Conditions

| | | MIN | NOM | MAX | UNIT |
|--------------|---------------------------------------|-------------------|----------|----------|------|
| V_{CC} | Supply voltage | 4.5 | 5 | 5.5 | V |
| V_{IH} | High-level input voltage (T1IN, T2IN) | 2 | | | V |
| V_{IL} | Low-level input voltage (T1IN, T2IN) | | | 0.8 | V |
| $R1IN, R2IN$ | Receiver input voltage | | | ±30 | V |
| T_A | Operating free-air temperature | MAX232 MAX232I | 0 -40 | 70 55 | °C |

7.4 Thermal Information

| THERMAL METRIC(1) | MAX232S | MAX232DSW | MAX232M | MAX232MS | UNIT | |
|-------------------|--|-----------|---------|----------|------|------|
| | SOIC | SOIC wide | POP | SOP | | |
| | 16 PINS | 16 PINS | 16 PINS | 16 PINS | | |
| R_{JA} | Junction-to-ambient thermal resistance | 73 | 57 | 57 | 64 | °C/W |

(1) For more information about traditional and new thermal metrics, see the IC Package Thermal Metrics application report (SPRA063).

7.5 Electrical Characteristics — Device

over recommended range of supply voltage and operating free-air temperature (unless otherwise noted) (see Figure 5)

| PARAMETER | TEST CONDITIONS ⁽¹⁾ | MIN | TYP ⁽²⁾ | MAX | UNIT |
|-----------|--------------------------------|---|--------------------|-----|------|
| I_{CC} | Supply current | $V_{CC} = 5.5$ V, all outputs open, $T_A = 25$ °C | 9 | 10 | mA |

(1) Test conditions are C1-C4 = 1 μ F at $V_{CC} = 5$ V \pm 0.5 V.

(2) All typical values are at $V_{CC} = 5$ V and $T_A = 25$ °C.

7.6 Electrical Characteristics — Driver

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

| PARAMETER | TEST CONDITIONS ⁽¹⁾ | MIN | TYP ⁽²⁾ | MAX | UNIT |
|--|---|-----|--------------------|-----|---------------|
| V_{OH} High-level output voltage | T1OUT, T2OUT $R_L = 3\text{ k}\Omega$ to GND | | 5 | 7 | V |
| V_{OL} Low-level output voltage ⁽³⁾ | T1OUT, T2OUT $R_L = 3\text{ k}\Omega$ to GND | | –7 | –5 | V |
| R_D Output resistance | T1OUT, T2OUT $V_{OL} = V_{OH} = 0$, $V_{CC} = 5\text{ V}$ | 300 | | | Ω |
| I_{OZ} ⁽⁴⁾ Short-circuit output current | T1OUT, T2OUT $V_{CC} = 5.5\text{ V}$, $V_O = 0\text{ V}$ | | ± 10 | | mA |
| I_{IH} Short-circuit input current | T1IN, T2IN $V_I = 0$ | | | 200 | μA |

(1) Test conditions are C1–C4 = $1\text{ }\mu\text{F}$ at $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$.

(2) All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$.

(3) The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

(4) Not more than one output should be shorted at a time.

7.7 Electrical Characteristics — Receiver

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

| PARAMETER | TEST CONDITIONS ⁽¹⁾ | MIN | TYP ⁽²⁾ | MAX | UNIT |
|--|--|-----|--------------------|-----|------------|
| V_{OH} High-level output voltage | R1OUT, R2OUT $I_{OL} = -1\text{ mA}$ | 3.5 | | | V |
| V_{OL} Low-level output voltage ⁽³⁾ | R1OUT, R2OUT $I_{OL} = 3.2\text{ mA}$ | | | 0.4 | V |
| V_{TH} Receiver positive-going input threshold voltage | R1IN, R2IN $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ | | 1.7 | 2.4 | V |
| V_{TL} Receiver negative-going input threshold voltage | R1IN, R2IN $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ | 0.8 | 1.2 | | V |
| V_{IL} Input hysteresis voltage | R1IN, R2IN $V_{CC} = 5\text{ V}$ | 0.2 | 0.5 | 1 | V |
| R_i Receiver input resistance | R1IN, R2IN $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$ | 3 | 5 | 7 | k Ω |

(1) Test conditions are C1–C4 = $1\text{ }\mu\text{F}$ at $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$.

(2) All typical values are at $V_{CC} = 5\text{ V}$, $T_A = 25^\circ\text{C}$.

(3) The algebraic convention, in which the least-positive (most negative) value is designated minimum, is used in this data sheet for logic voltage levels only.

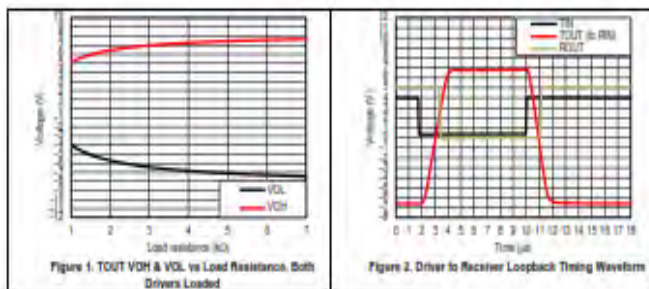
7.8 Switching Characteristics

over recommended ranges of supply voltage and operating free-air temperature (unless otherwise noted)

| PARAMETER | TEST CONDITIONS ⁽¹⁾ | MIN | TYP ⁽²⁾ | MAX | UNIT |
|--|---|-----|--------------------|-----|------------------|
| SR Driver slew rate | $R_L = 3\text{ k}\Omega$ to $7\text{ k}\Omega$, see Figure 4 | | | 30 | V/ μs |
| SPN Driver transition region slew rate | see Figure 5 | | 3 | | V/ μs |
| Data rate | See TOUT switching | | 120 | | kb/s |
| t_{PLH} Receiver propagation delay time, low- to high-level output | TTL load, see Figure 3 | | 500 | | ns |
| t_{PHL} Receiver propagation delay time, high- to low-level output | TTL load, see Figure 3 | | 500 | | ns |

(1) Test conditions are C1–C4 = $1\text{ }\mu\text{F}$ at $V_{CC} = 5\text{ V} \pm 0.5\text{ V}$.

7.9 Typical Characteristics



3. Datasheet IC D1307



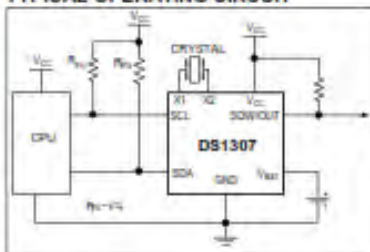
DS1307

64 x 8, Serial, I²C Real-Time Clock

GENERAL DESCRIPTION

The DS1307 serial real-time clock (RTC) is a low-power, full binary-coded decimal (BCD) clock/calendar plus 56 bytes of NV SRAM. Address and data are transferred serially through an I²C, bidirectional bus. The clock/calendar provides seconds, minutes, hours, day, date, month, and year information. The end of the month date is automatically adjusted for months with fewer than 31 days, including corrections for leap year. The clock operates in either the 24-hour or 12-hour format with AM/PM indicator. The DS1307 has a built-in power-sense circuit that detects power failures and automatically switches to the backup supply. Timekeeping operation continues while the part operates from the backup supply.

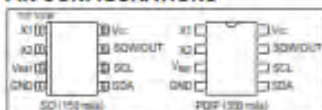
TYPICAL OPERATING CIRCUIT



FEATURES

- Real-Time Clock (RTC) Counts Seconds, Minutes, Hours, Date of the Month, Month, Day of the week, and Year with Leap-Year Compensation Valid Up to 2100
- 56-Byte, Battery-Backed, General-Purpose RAM with Unlimited Writes
- I²C Serial Interface
- Programmable Square-Wave Output Signal
- Automatic Power-Fail Detect and Switch Circuitry
- Consumes Less than 500nA in Battery-Backup Mode with Oscillator Running
- Optional Industrial Temperature Range: -40°C to +85°C
- Available in 5-Pin Plastic DIP or SO
- Underwriters Laboratories (UL) Recognized

PIN CONFIGURATIONS



ORDERING INFORMATION

| PART | TEMP RANGE | VOLTAGE (V) | PIN-PACKAGE | TOP MARK* |
|--------------|----------------|-------------|-------------------------------|-----------|
| DS1307+ | 0°C to +70°C | 5.0 | 5 PDIP (300 mils) | DS1307 |
| DS1307N+ | -40°C to +85°C | 5.0 | 5 PDIP (300 mils) | DS1307N |
| DS1307Z+ | 0°C to +70°C | 5.0 | 5 SO (150 mils) | DS1307 |
| DS1307ZN+ | -40°C to +85°C | 5.0 | 5 SO (150 mils) | DS1307N |
| DS1307Z+T&R | 0°C to +70°C | 5.0 | 5 SO (150 mils) Tape and Reel | DS1307 |
| DS1307ZN+T&R | -40°C to +85°C | 5.0 | 5 SO (150 mils) Tape and Reel | DS1307N |

* Denotes a lead-free/Pb-free compliant package.

* A "+" anywhere on the top mark indicates a lead-free package. An "N" anywhere on the top mark indicates an industrial temperature range device.

ABSOLUTE MAXIMUM RATINGS

| | |
|---|---|
| Voltage Range on Any Pin Relative to Ground | -0.5V to +7.0V |
| Operating Temperature Range (Noncondensing) | |
| Commercial | 0°C to +70°C |
| Industrial | -40°C to +85°C |
| Storage Temperature Range | -55°C to +125°C |
| Soldering Temperature (DIP, leads) | +260°C for 10 seconds |
| Soldering Temperature (surface mount) | Refer to the JPC/JEDEC J-STD-020 Specification. |

Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. These are stress ratings only; and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of the specifications is not implied. Exposure to the absolute maximum rating conditions for extended periods may affect device reliability.

RECOMMENDED DC OPERATING CONDITIONS(T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|----------------------------------|------------------|------------|------|-----|-----------------------|-------|
| Supply Voltage | V _{CC} | | 4.5 | 5.0 | 5.5 | V |
| Logic 1 Input | V _{HI} | | 2.2 | | V _{CC} + 0.3 | V |
| Logic 0 Input | V _{LO} | | -0.3 | | +0.8 | V |
| V _{BAT} Battery Voltage | V _{BAT} | | 2.0 | 3 | 3.5 | V |

DC ELECTRICAL CHARACTERISTICS(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|--|------------------------|------------|-----------------------------|----------------------------|-----------------------------|-------|
| Input Leakage (SCL) | I _{IL} | | -1 | | 1 | μA |
| I/O Leakage (SDA, SQW/OUT) | I _{I/O} | | -1 | | 1 | μA |
| Logic 0 Output (I _{OL} = 5mA) | V _{OL} | | | | 0.4 | V |
| Active Supply Current (f _{OSC} = 100kHz) | I _{CC(A)} | | | | 1.5 | mA |
| Standby Current | I _{CC(S)} | (Note 3) | | | 200 | μA |
| V _{BAT} Leakage Current | I _{BAT(LEAK)} | | | 5 | 50 | nA |
| Power-Fall Voltage (V _{BAT} = 3.0V) | V _{PF} | | 1.216 x V _{BAT} | 1.25 x V _{BAT} | 1.264 x V _{BAT} | V |

DC ELECTRICAL CHARACTERISTICS(V_{CC} = 0V, V_{BAT} = 3.0V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.) (Notes 1, 2)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|---|---------------------|------------|-----|-----|-----|-------|
| V _{BAT} Current (OSC ON); SQW/OUT OFF | I _{BAT1} | | | 300 | 500 | nA |
| V _{BAT} Current (OSC ON); SQW/OUT ON (32kHz) | I _{BAT2} | | | 450 | 500 | nA |
| V _{BAT} Data-Retention Current (Oscillator Off) | I _{BATDR1} | | | 10 | 100 | nA |

WARNING: Negative undershoots below -0.3V while the part is in battery-backed mode may cause loss of data.

AC ELECTRICAL CHARACTERISTICS(V_{CC} = 4.5V to 5.5V; T_A = 0°C to +70°C, T_A = -40°C to +85°C.)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|--|----------------------|--------------|-----|-----|------|-------|
| SCL Clock Frequency | f _{SCL} | | 0 | | 100 | kHz |
| Bus Free Time Between a STOP and START Condition | t _{BUS} | | 4.7 | | | μs |
| Hold Time (Repeated) START Condition | t _{HD,STA} | (Note 4) | 4.0 | | | μs |
| LOW Period of SCL Clock | t _{LOW} | | 4.7 | | | μs |
| HIGH Period of SCL Clock | t _{HIGH} | | 4.0 | | | μs |
| Setup Time for a Repeated START Condition | t _{SD,STA} | | 4.7 | | | μs |
| Data Hold Time | t _{HD,DATA} | | 0 | | | μs |
| Data Setup Time | t _{SD,DATA} | (Notes 5, 6) | 250 | | | ns |
| Rise Time of Both SDA and SCL Signals | t _r | | | | 1000 | ns |
| Fall Time of Both SDA and SCL Signals | t _f | | | | 300 | ns |
| Setup Time for STOP Condition | t _{SD,STOP} | | 4.7 | | | μs |

CAPACITANCE(T_A = +25°C)

| PARAMETER | SYMBOL | CONDITIONS | MIN | TYP | MAX | UNITS |
|------------------------------------|-----------------|------------|-----|-----|-----|-------|
| Pin Capacitance (SDA, SCL) | C _{IO} | | | | 10 | pF |
| Capacitance Load for Each Bus Line | C _L | (Note 7) | | | 400 | pF |

Note 1: All voltages are referenced to ground.**Note 2:** Limits at -40°C are guaranteed by design and are not production tested.**Note 3:** I_{OL} specified with V_{CC} = 5.0V and SDA, SCL = 5.0V.**Note 4:** After this period, the first clock pulse is generated.**Note 5:** A device must internally provide a hold time of at least 350ns for the SDA signal (referred to the V_{IL(SCL)} of the SCL signal) to bridge the undefined region of the falling edge of SCL.**Note 6:** The maximum t_{SD,DATA} only has to be met if the device does not stretch the LOW period (t_{LOW}) of the SCL signal.**Note 7:** C_L—total capacitance of one bus line in pF.

4. Datasheet IC PCF 8574

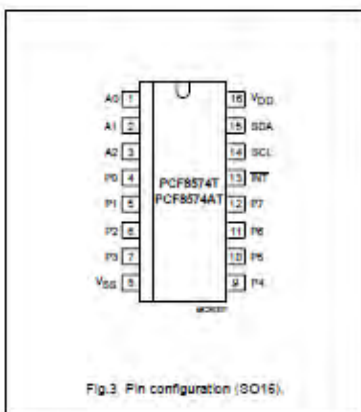
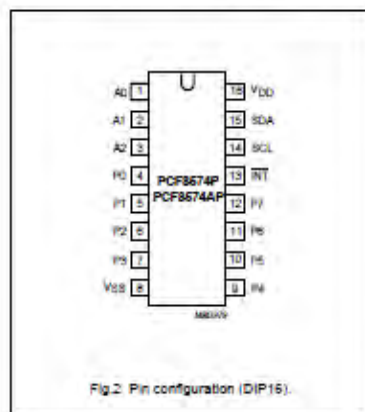
Remote 8-bit I/O expander for I²C-bus

PCF8574

5 PINNING

5.1 DIP16 and SO16 packages

| SYMBOL | PIN | DESCRIPTION |
|-----------------|-----|-------------------------------|
| A0 | 1 | address input 0 |
| A1 | 2 | address input 1 |
| A2 | 3 | address input 2 |
| P0 | 4 | quasi-bidirectional I/O 0 |
| P1 | 5 | quasi-bidirectional I/O 1 |
| P2 | 6 | quasi-bidirectional I/O 2 |
| P3 | 7 | quasi-bidirectional I/O 3 |
| V _{SS} | 8 | supply ground |
| P4 | 9 | quasi-bidirectional I/O 4 |
| P5 | 10 | quasi-bidirectional I/O 5 |
| P6 | 11 | quasi-bidirectional I/O 6 |
| P7 | 12 | quasi-bidirectional I/O 7 |
| INT | 13 | Interrupt output (active LOW) |
| SCL | 14 | serial clock line |
| SDA | 15 | serial data line |
| V _{DD} | 16 | supply voltage |



Remote 8-bit I/O expander for I²C-bus

PCF8574

6 CHARACTERISTICS OF THE I²C-BUS

The I²C-bus is for 2-way, 2-line communication between different ICs or modules. The two lines are a serial data line (SDA) and a serial clock line (SCL). Both lines must be connected to a positive supply via a pull-up resistor when connected to the output stages of a device. Data transfer may be initiated only when the bus is not busy.

6.1 Bit transfer

One data bit is transferred during each clock pulse. The data on the SDA line must remain stable during the HIGH period of the clock pulse as changes in the data line at this time will be interpreted as control signals (see Fig.5).

6.2 Start and stop conditions

Both data and clock lines remain HIGH when the bus is not busy. A HIGH-to-LOW transition of the data line, while the clock is HIGH is defined as the start condition (S). A LOW-to-HIGH transition of the data line while the clock is HIGH is defined as the stop condition (P) (see Fig.6).

6.3 System configuration

A device generating a message is a 'transmitter', a device receiving is the 'receiver'. The device that controls the message is the 'master' and the devices which are controlled by the master are the 'slaves' (see Fig.7).

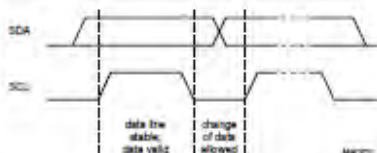


Fig.5 Bit transfer.

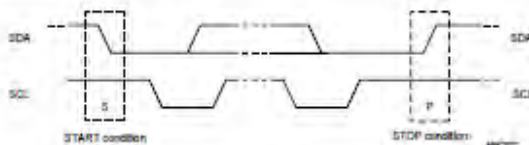


Fig.6 Definition of start and stop conditions.

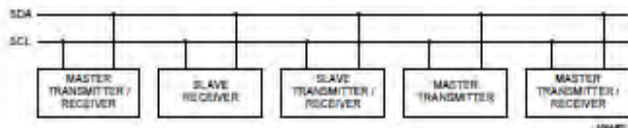


Fig.7 System configuration.

Remote 8-bit I/O expander for I²C-bus

PCF8574

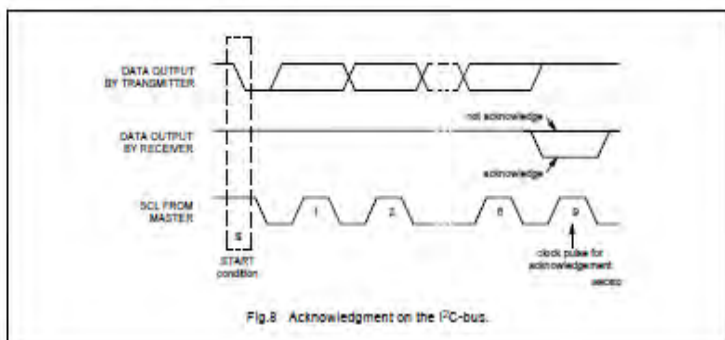
6.4 Acknowledge

The number of data bytes transferred between the start and the stop conditions from transmitter to receiver is not limited. Each byte of eight bits is followed by one acknowledge bit (see Fig. 8). The acknowledge bit is a HIGH level put on the bus by the transmitter whereas the master generates an extra acknowledge related clock pulse.

A slave receiver which is addressed must generate an acknowledge after the reception of each byte. Also a master must generate an acknowledge after the reception

of each byte that has been clocked out of the slave transmitter. The device that acknowledges has to pull down the SDA line during the acknowledge clock pulse, so that the SDA line is stable LOW during the HIGH period of the acknowledge related clock pulse, set-up and hold times must be taken into account.

A master receiver must signal an end of data to the transmitter by not generating an acknowledge on the last byte that has been clocked out of the slave. In this event the transmitter must leave the data line HIGH to enable the master to generate a stop condition.



5. Datasheet LCD 16x2

www.p-tec.net
sales@p-tec.net



Tel: (714) 589 3122
Fax: (714) 589 3562

| PC1602A-L (16x2) Character LCD Display | | | | | |
|--|--|-----------------------------|-----------------------|-----------------------|------------------------------------|
| Features *16 Character, 2 Line *View Angle 12H or 8H *TN or STN Final *Extended Temperature Range available *Several Character Types available *LED or EL Backlight available | Absolute Maximum Ratings at $T_A = 25^{\circ}\text{C}$ | | | | |
| | Item | Symbol | Min | Max | Unit |
| | Power Supply (Logic) | V _{dd} | -0.5 | 7.0 | V |
| | Power Supply (LCD) | V _o | V _{dd} - 0.5 | V _{dd} + 2.0 | V |
| | Input Voltage | V _i | -0.5 | V _{dd} | V |
| | Operating Temperature (Standard) | T _{opr} | 0 | 50 | °C |
| | Storage Temperature (Standard) | T _{stg} | -20 | 70 | °C |
| | Extended Operating Temperature | T _{opr} | -20 | 70 | °C |
| | Extended Storage Temperature | T _{stg} | -50 | 80 | °C |
| Electrical Characteristics at $T_A = 25^{\circ}\text{C}$, V _{dd} =5V±0.25V | | | | | |
| Item | Symbol | Min | Typ | Max | Unit |
| Power Supply (Logic) | V _{dd} -V _{ss} | 4.7 | 5.0 | 5.3 | V |
| Supply Current | I _{dd} | — | 2.0 | 3.0 | mA |
| LCD Driving Voltage | V _{dd} -V _o | 4.2 | 4.5 | 4.8 | V |
| Input Voltage "H" | V _{ih} | 2.3 | — | V _{dd} | V |
| Input Voltage "L" | V _{il} | — | — | 0.6 | V |
| Output Voltage "H" | V _{oh} | 2.4 | — | — | V |
| Output Voltage "L" | V _{ol} | — | — | 0.4 | V |
| Interface Pin Connection | | | | | |
| No | Symbol | Function | No | Symbol | Function |
| 1 | V _{ss} | Power Supply (GND) | 9 | DB2 | Data Bus Line 2 |
| 2 | V _{dd} | Power Supply (+5V) | 10 | DB3 | Data Bus Line 3 |
| 3 | V _o | Contrast Adjust | 11 | DB4 | Data Bus Line 4 |
| 4 | RS | Instruction/Register Select | 12 | DB5 | Data Bus Line 5 |
| 5 | R/W | Read/Write | 13 | DB6 | Data Bus Line 6 |
| 6 | E | Enable Signal | 14 | DB7 | Data Bus Line 7 |
| 7 | DB0 | Data Bus Line 0 | 15 | A | Power Supply for LED Backlight (+) |
| 8 | DB1 | Data Bus Line 1 | 16 | K | Power Supply for LED Backlight (-) |
| LED Backlight Specifications | | | | | |
| Forward Voltage | Forward Current | Power Dissipation | Peak Wavelength | | |
| 4.2V (Typ) | 100mA (Max) | 900mW (Max) | 570nm (Typ) | | |
| Mechanical Specifications | | | | | |
| Item | Overall Size | Viewing Area | Character Size | Character Pitch | Dot Size |
| Specifications | 90.0W x 36.0H x 8.0T (BL 12.7T) | 65.0W x 16.0H | 2.00W x 5.56H | 3.55W x 8.94H | 0.56W x 0.60H |
| All dimensions are in mm. Tolerance is ±0.1 unless otherwise specified. | | | | | |

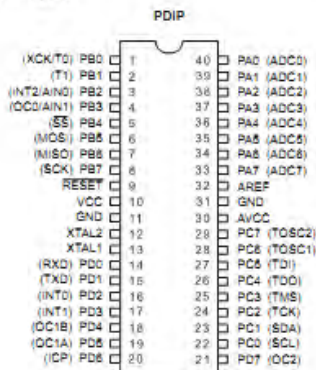
LAMPIRAN 3

Features

- High-performance, Low-power AVR® 8-bit Microcontroller
- Advanced RISC Architecture
 - 131 Powerful Instructions – Most Single-clock Cycle Execution
 - 32 x 8 General Purpose Working Registers
 - Fully Static Operation
 - Up to 16 MIPS Throughput at 16 MHz
 - On-chip 2-cycle Multiplier
- Nonvolatile Program and Data Memories
 - 16K Bytes of In-System Self-Programmable Flash
 - Endurance: 10,000 Write/Erase Cycles
 - Optional Boot Code Section with Independent Lock Bits
 - In-System Programming by On-chip Boot Program
 - True Read-While-Write Operation
 - 512 Bytes EEPROM
 - Endurance: 100,000 Write/Erase Cycles
 - 1K Byte Internal SRAM
 - Programming Lock for Software Security
- JTAG (IEEE std. 1149.1 Compliant) Interface
 - Boundary-scan Capabilities According to the JTAG Standard
 - Extensive On-chip Debug Support
 - Programming of Flash, EEPROM, Fuses, and Lock Bits through the JTAG interface
- Peripheral Features
 - Two 8-bit Timer/Counters with Separate Prescalers and Compare Modes
 - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
 - Real Time Counter with Separate Oscillator
 - Four PWM Channels
 - 8-channel, 10-bit ADC
 - 8 Single-ended Channels
 - 7 Differential Channels in TQFP Package Only
 - 2 Differential Channels with Programmable Gain at 1x, 10x, or 200x
 - Byte-oriented Two-wire Serial Interface
 - Programmable Serial USART
 - Master/Slave SPI Serial Interface
 - Programmable Watchdog Timer with Separate On-chip Oscillator
 - On-chip Analog Comparator
- Special Microcontroller Features
 - Power-on Reset and Programmable Brown-out Detection
 - Internal Calibrated RC Oscillator
 - External and Internal Interrupt Sources
 - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby and Extended Standby
- I/O and Packages
 - 32 Programmable I/O Lines
 - 40-pin PDIP, 44-lead TQFP, and 44-pad MLF
- Operating Voltages
 - 2.7 - 5.5V for ATmega16L
 - 4.5 - 5.5V for ATmega16
- Speed Grades
 - 0 - 8 MHz for ATmega16L
 - 0 - 16 MHz for ATmega16

Pin Configurations

Figure 1. Pinouts ATmega16



Pin Descriptions

VCC

Digital supply voltage.

GND

Ground.

Port A (PA7..PA0)

Port A serves as the analog inputs to the A/D Converter.

Port A also serves as an 8-bit bi-directional I/O port, if the A/D Converter is not used. Port pins can provide internal pull-up resistors (selected for each bit). The Port A output buffers have symmetrical drive characteristics with both high sink and source capabilities. When pins PA0 to PA7 are used as inputs and are externally pulled low, they will sink current if the internal pull-up resistors are activated. The Port A pins are tri-stated when a reset condition becomes active, even if the clock is not running.

| | |
|--------------------------|---|
| Port B (PB7..PB0) | <p>Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port B also serves the functions of various special features of the ATmega16 as listed on page 55.</p> |
| Port C (PC7..PC0) | <p>Port C is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port C output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running. If the JTAG interface is enabled, the pull-up resistors on pins PC5(TDI), PC3(TMS) and PC2(TCK) will be activated even if a reset occurs.</p> <p>Port C also serves the functions of the JTAG interface and other special features of the ATmega16 as listed on page 58.</p> |
| Port D (PD7..PD0) | <p>Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.</p> <p>Port D also serves the functions of various special features of the ATmega16 as listed on page 60.</p> |
| RESET | <p>Reset Input. A low level on this pin for longer than the minimum pulse length will generate a reset, even if the clock is not running. The minimum pulse length is given in Table 15 on page 35. Shorter pulses are not guaranteed to generate a reset.</p> |
| XTAL1 | Input to the inverting Oscillator amplifier and input to the internal clock operating circuit. |
| XTAL2 | Output from the inverting Oscillator amplifier. |
| AVCC | AVCC is the supply voltage pin for Port A and the A/D Converter. It should be externally connected to V_{CC} , even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. |
| AREF | AREF is the analog reference pin for the A/D Converter. |

Status Register

The Status Register contains information about the result of the most recently executed arithmetic instruction. This information can be used for altering program flow in order to perform conditional operations. Note that the Status Register is updated after all ALU operations, as specified in the Instruction Set Reference. This will in many cases remove the need for using the dedicated compare instructions, resulting in faster and more compact code.

The Status Register is not automatically stored when entering an interrupt routine and restored when returning from an interrupt. This must be handled by software.

The AVR Status Register – SREG – is defined as:

| Bit | 7 | 6 | 5 | 4 | 3 | 2 | 1 | 0 | |
|---------------|-----|-----|-----|-----|-----|-----|-----|-----|------|
| | I | T | H | S | V | N | Z | C | SREG |
| Read/Write | R/W | R/W | R/W | R/W | R/W | R/W | R/W | R/W | |
| Initial Value | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | |

- **Bit 7 – I: Global Interrupt Enable**

The Global Interrupt Enable bit must be set for the interrupts to be enabled. The individual interrupt enable control is then performed in separate control registers. If the Global Interrupt Enable Register is cleared, none of the interrupts are enabled independent of the individual interrupt enable settings. The I-bit is cleared by hardware after an interrupt has occurred, and is set by the RETI instruction to enable subsequent interrupts. The I-bit can also be set and cleared by the application with the SEI and CLI instructions, as described in the instruction set reference.

- **Bit 6 – T: Bit Copy Storage**

The Bit Copy instructions BLD (Bit Load) and BST (Bit Store) use the T-bit as source or destination for the operated bit. A bit from a register in the Register file can be copied into T by the BST instruction, and a bit in T can be copied into a bit in a register in the Register file by the BLD instruction.

- **Bit 5 – H: Half Carry Flag**

The Half Carry Flag H indicates a half carry in some arithmetic operations. Half Carry is useful in BCD arithmetic. See the "Instruction Set Description" for detailed information.

- **Bit 4 – S: Sign Bit, $S = N \oplus V$**

The S-bit is always an exclusive or between the negative flag N and the two's complement overflow flag V. See the "Instruction Set Description" for detailed information.

- **Bit 3 – V: Two's Complement Overflow Flag**

The Two's Complement Overflow Flag V supports two's complement arithmetics. See the "Instruction Set Description" for detailed information.

- **Bit 2 – N: Negative Flag**

The Negative Flag N indicates a negative result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 1 – Z: Zero Flag**

The Zero Flag Z indicates a zero result in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

- **Bit 0 – C: Carry Flag**

The Carry Flag C indicates a carry in an arithmetic or logic operation. See the "Instruction Set Description" for detailed information.

Panduan R-Water

1. Fungsi Tombol

- Angka 1-9 sebagai input angka pada LCD
- Huruf A untuk fungsi input nominal rupiah
- Huruf B untuk input nominal Litter
- Huruf C untuk fungsi counter nominal Rp dengan litter
- Huruf D untuk menkonfirmasi data yang telah di inputkan dengan menyimpan data tersebut
- Tanda # sebagai fungsi input data harga/litter untuk air bersih gratis dan air bersih berbayar
- Tanda * untuk cek id RFID CARD



2. Cara menggunakan RFID-SIM :

a. Mengatur harga

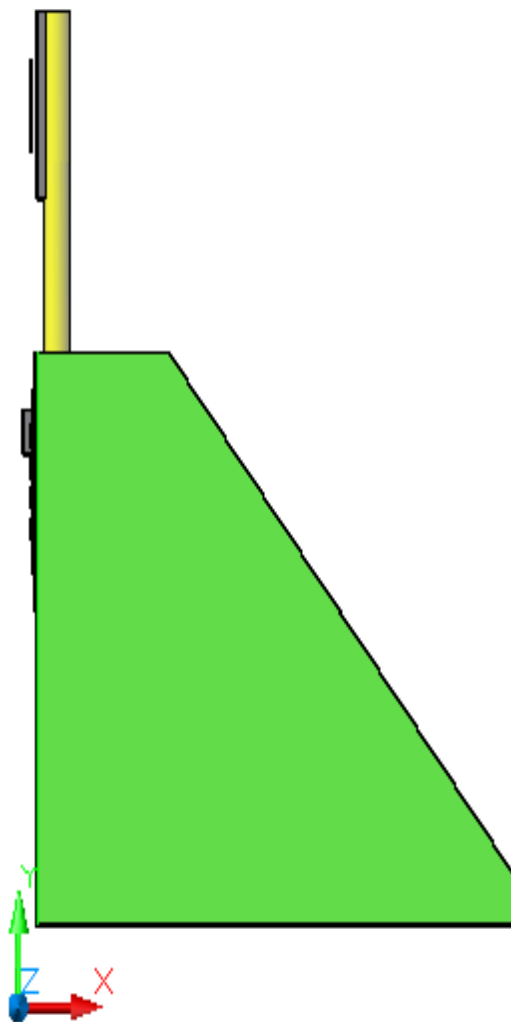
- i. Tekan tombol # untuk menampilkan pengaturan Harga subsidi dan non subsidi dalam rupiah.
- ii. Tekan tombol huruf A untuk setting harga air bersih
- iii. Tekan tombol huruf B untuk setting harga air bersih berbayar.
- iv. Tekan tombol huruf D untuk save data inputan yang di inginkan.

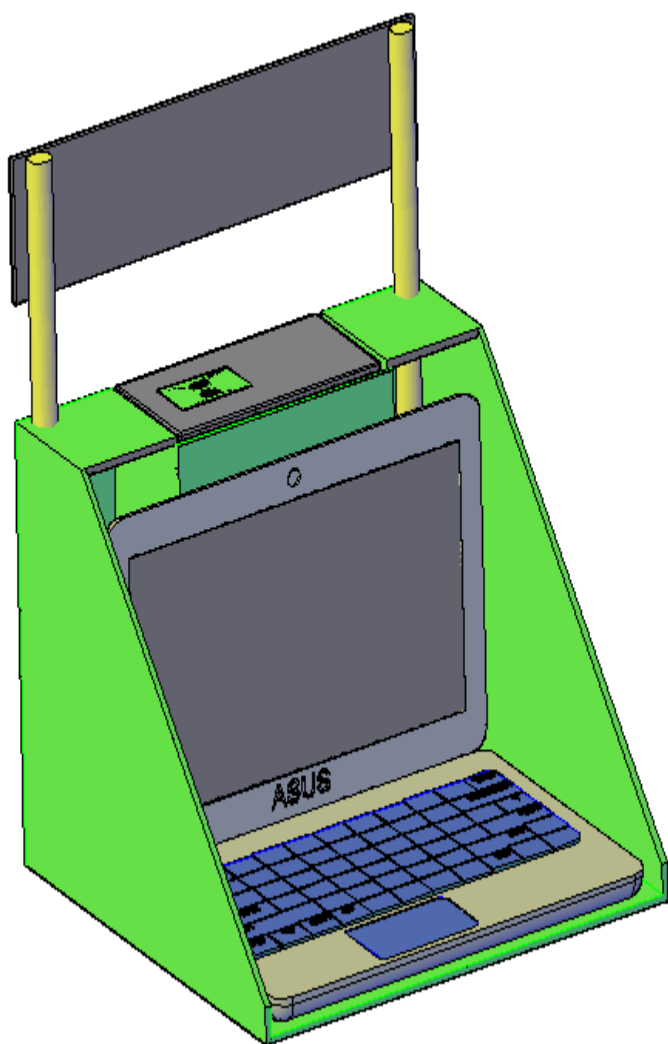
b. Mengecek ID pada RFID card.

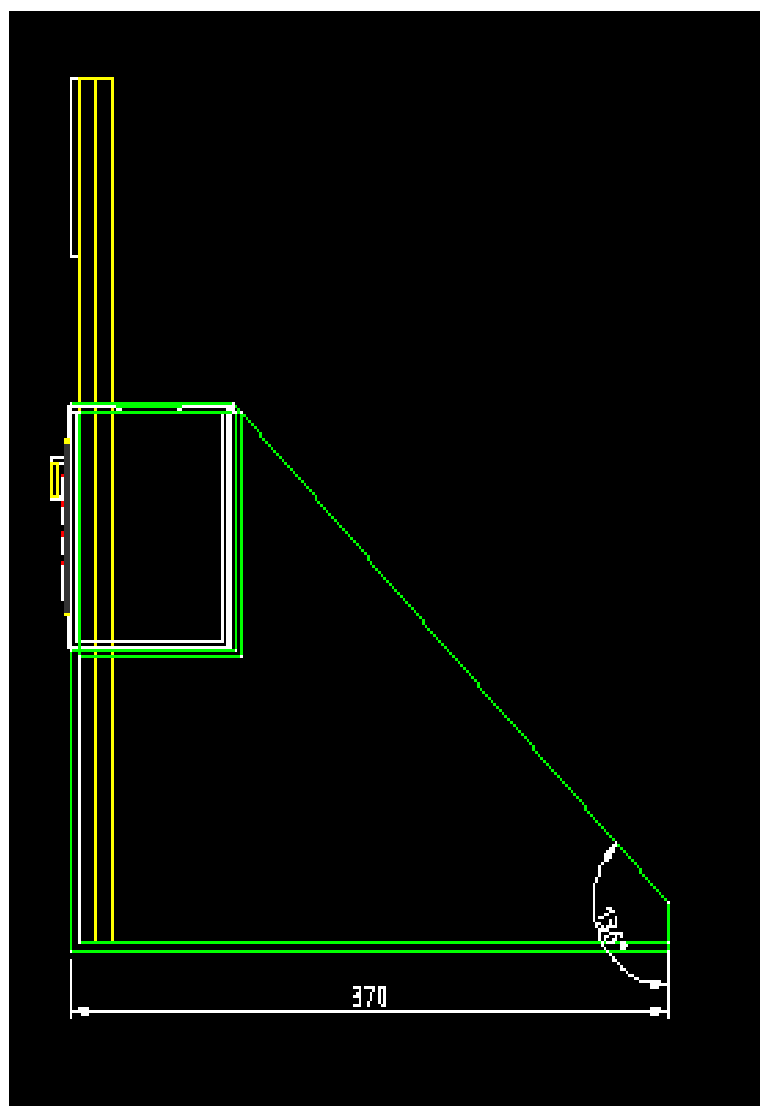
- i. Tekan tombol * untuk menampilkan ID yang akan di inputkan melalui tag RFID reader

- ii. Tempelkan kartu RFID kemudian tag RFID reader untuk menampilkan ID kartu yang akan di inputkan untuk pembuatan RFID SIM
 - iii. ID yang tampil akan di inputkan untuk mendaftarkan pembuatan RFID SIM
- c. Menjalankan RFID SIM untuk pembatasan subsidi BBM
 - i. Tag R-SIM ke RFID reader untuk memulai pembelian BBM bersubsidi
 - ii. Tekan tombol huruf A untuk menampilkan pembelian dengan nominal rupiah
 - iii. Tekan tombol huruf B untuk menampilkan pembelian dengan nominal liter
 - iv. Tekan tombol D untuk mengkonfirmasi nominal pembelian
 - v. Tekan tombol huruf C untuk mengaktifkan counter perhitungan rupiah dan liter
 - vi. Sisa subsidi akan di tampilkan pada lcd
- d. Tombol reset
 - i. Pada mikro terdapat tombol reset untuk mendapatkan kembali air bersih pada hari berikutnya

Desain Hardware :

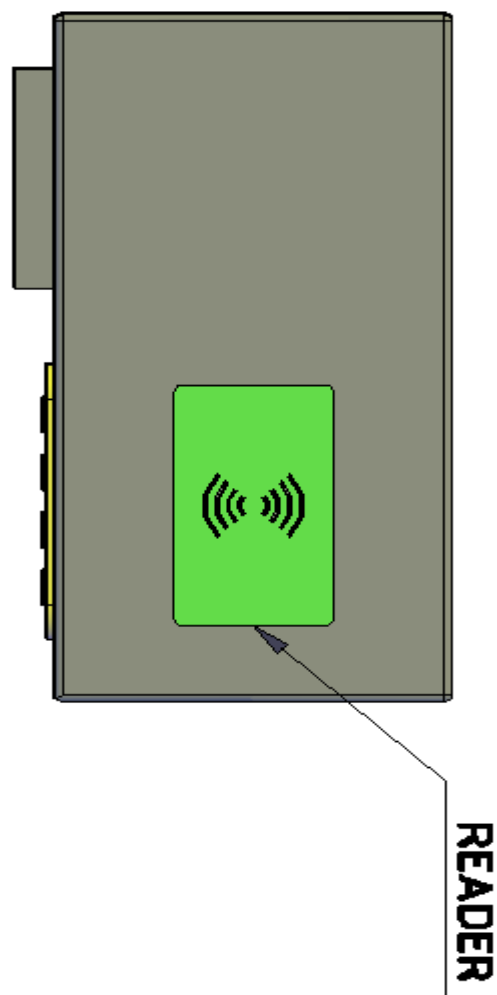


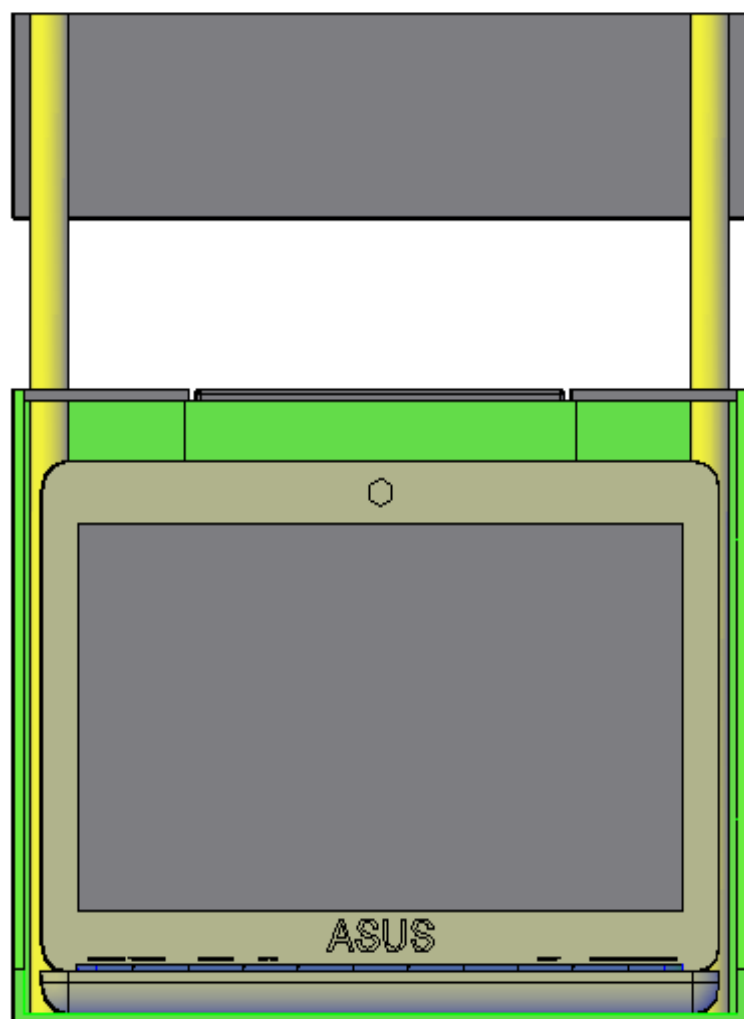




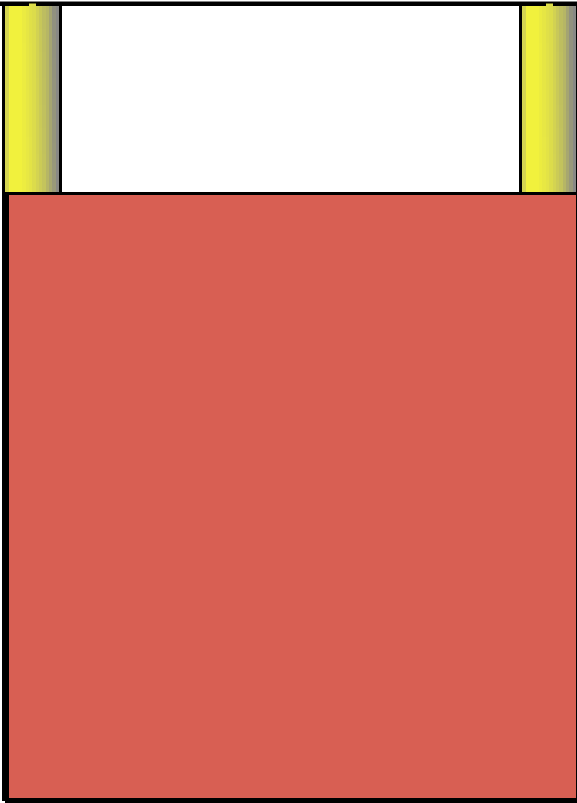








TANDON



DAFTAR PUSTAKA

- [1] Budi,Ronald, *Programming With Microsoft Visual Basic, Skripta*,April 2010.
- [2] Winoto, Ardi. *Mikrokontroler ATmega8/32/16/8535 dan Pemrogramannya dengan Bahasa C*. Bandung: Informatika. 2008.
- [3] Ilie-ZudorElisabeth, KemenySolt, EgriPéter, Monostory László. *The RFID Technology and its current Applications*. 2006.
- [4] Adiyaksa, Baskara, *Sistem Mudah Belajar Elektronika* , Yogyakarta, Andi Yogyakarta, Maret 2010.

RIWAYAT HIDUP PENULIS



Nama : Gamal Armando
TTL : Surabaya, 03 Februari 1993
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat Rumah : Jln. Rungkut Asri
Tengah 1 No.12
Surabaya
Telp/HP : 082230007793
E-mail : gamalarmando@yahoo.com
Hobi : Otomotif

RIWAYAT PENDIDIKAN

- © 1999 – 2005 : SD Muhammadiyah 4 Surabaya
- © 2005 – 2008 : SMP Negeri 12 Surabaya
- © 2008 – 2011 : SMA Negeri 2 Surabaya
- © 2011 – sekarang : Bidang Studi Komputer Kontrol, Program D3 Teknik Elektro, ITS

PENGALAMAN KERJA

- © Kerja Praktek di bagian Elektronika Bandara di Angkasa Pura I Bandara Juanda Surabaya.

PENGALAMAN ORGANISASI

- © Staf Departemen Dewan Perwakilan Mahasiswa Jurusan Himpunan Mahasiswa D3 Teknik Elektro 2012 – 2013

{Halaman ini sengaja dikosongkan}

RIWAYAT HIDUP PENULIS



Nama : Muhammad Imam
Ainul Hidayat
TTL : Lamongan, 25 Mei
1992
Jenis Kelamin : Laki-laki
Agama : Islam
Alamat Rumah : Parengan Maduran
Lamongan RT 03 RW
03
Telp/HP : 082231998890
085645557878
E-mail : jazter_mvpl6@yahoo.com
Hobi : Olahraga

RIWAYAT PENDIDIKAN

- 📅 1999 – 2005 : MIM 1 Pangkatrejo
- 📅 2005 – 2008 : MTS Islam Al Mukmin
- 📅 2008 – 2011 : MA Al Mukmin
- 📅 2011 – sekarang : Bidang Studi Komputer Kontrol, Program D3
Teknik Elektro, ITS

PENGALAMAN KERJA

- 📅 Kerja Praktek di PT. Telkomsel Profinsi Jawa Timur
- 📅 Freelance PT. Absabat instalasi Viber optic

PENGALAMAN ORGANISASI

- 📅 Staf Departemen Kesejahteraan mahasiswa HIMA D3 Teknik
Elektro 2012 – 2013
- 📅 Kadept SyiarCBSO Salman Al Farisi Jurusan HIMA D3 Teknik
Elektro 2013 – 2014

{Halaman ini sengaja dikosongkan}